2006

Statistics on Stiefel manifolds

Gabriel Camano-Garcia
Iowa State University

Statistics on Stiefel manifolds

by

Gabriel Camano-Garcia

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Alicia Carriquiry, Co-major Professor
Wolfgang Kliemann, Co-major Professor
Stephen B. Vardeman
Dean L. Isaacson
Heike Hofmann

Iowa State University

Ames, Iowa

2006

Graduate College
Iowa State University

This is to certify that the Doctoral dissertation of

Gabriel Camano-Garcia

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

Co-major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank my major professors Dr. Alicia Carriquiry and Dr. Wolfgang Kliemann for all their support and encouragement. This thesis would not have been possible without their invaluable help and guidance. I would like to specially thanks Dr. Carriquiry for her infinite patience. I would also like to thank the members of my POS committee with special thanks to Dr. Stephen B. Vardeman for all his support throughout my time here at Iowa State. I would also like to thank Laura Nalbarte, Juan José Goyeneche, and the late Jorge Blanco to whom I will always be indebted for their support.

ABSTRACT

When the orientation of an object lies in a space of non-zero curvature usual distributions of probability cannot be used to describe its directions. One of such spaces is the Stiefel manifold. We focus on a probability distribution defined on that space, the matrix Langevin distribution. Classical and Bayesian methods of estimation of the parameter of the distribution are discussed. As the dimension of the Stiefel manifold increases, the more complicated the estimation process becomes given the complexity of the functions to be evaluated. A method is given that efficiently parameterizes the elements of the singular value decomposition of the parameter of the matrix Langevin distribution in terms of generalized Euler angles. How to implement that parameterization in the context of Bayesian estimation is shown. The methodology is illustrated with a dataset on trace element concentrations in bullet tips from the Federal Bureau of Investigations.

1 Introduction

1.1 Motivation

Often times, multiple measurements are collected on a sample of units. Consider, for example, a nationwide food consumption survey where individuals are asked about consumption of various food groups, or the laboratory analysis of bullet lead, where the concentration of trace elements in the lead alloy is used to obtain a chemical 'fingerprint' for each specimen. In many cases, it is of interest to group the sample of units of size N into 'similar' sub-groups, perhaps by implementing a hierarchical clustering algorithm on the n-dimensional multivariate measurements. In this case, the joint density of the sample units can be written down as a mixture of $n-$dimensional densities. The number d of components in the mixture is often assumed to be known, and component densities are sometimes assumed to be multivariate normal. Many clustering approaches have been proposed in the literature ([4],[15],[16], [27]) and we do not discuss them further.

We note, however, that some algorithms are flexible in that they allow identification of clusters with different shape, orientation and volume in $n-$dimensional space. An example is the model-based clustering algorithm proposed by [4]. For some families of distributions including the multivariate normal family, the eigenvalues and eigenvectors

of the component covariance matrices Σ_j, $j = 1, ..., d$ determine the shape, orientation and volume of the observation vectors in the jth group. Therefore, by writing Σ_j in its spectral form, it is possible to fit a mixture model where the assumption of homogeneous covariance matrices can be relaxed without significantly increasing the number of parameters that need to be estimated. Other non-model-based hierarchical clustering algorithms can result in groupings with heterogeneous within-group covariance matrices as well. If, in fact, the N sample vector observations can best be described by a mixture probability model where the component covariance matrices cannot be reasonably assumed to be homogeneous across groups, drawing inferences about mean differences across the groups can be challenging ([27]).

One approach to changing the orientation of the jth group of $n-$dimensional observation vectors X_{kj}, $k = 1, ..., d_j$ with d_j the number of observations in the jth group is to rotate the observation vectors by multiplying them by an orthogonal matrix T_j. In fact, by estimating a suitable set of d such orthogonal matrices $T_1, ..., T_d$ from a specific sample, it is possible to orient all m sample clusters in the same direction in $n-$dimensional space. In this case, the assumption of homogeneous orientation in the model for the transformed observation vectors $T_j X_{kj}$

$$f(T_j X_{kj}) = \sum_{j=1}^{d} f_j(T_j X_{jk} | T_j \mu_j, T_j \Sigma_j T_j')$$

can be justified. Here, $T_j \mu_j$ and $T_j \Sigma_j T_j'$ denote the mean vector and the covariance matrix of the observation vectors in the jth cluster after observation vectors have been rotated.

In this dissertation, we focus on the orthogonal matrices T_j and investigate plau-

sible probability models for the general class of special orthogonal matrices. We first define and describe properties of the Stiefel manifold, the appropriate class of compact manifolds that includes the special orthogonal group and discuss probability models on the Stiefel manifold. We then review and extend results on maximum likelihood (ML) estimation of model parameters on the Stiefel manifold and prove some of the properties of MLEs. A Bayesian approach for estimation of model parameters is proposed, and various algorithms for implementing the approach are described.

1.2 A specific example

The specific example that motivated this work arose from a research project carried out at Iowa State University in cooperation with the Federal Bureau of Investigations ([7]). The question of interest was whether bullets can be grouped into unique compositional groups based on the concentration of the trace elements silver, bismuth, copper, antimony and arsenic in the bullet lead alloy. If, in fact, bullets have a unique chemical 'fingerprint' or if compositional groups are small enough, then the probative value of trace element evidence in court would be high.

The data in the study consisted of a sample of 800 .38-caliber cartridges loaded with 158 grain, round nose bullets manufactured by the four major US bullet manufacturers: Cascade, Remington, Federal, and Winchester. Two hundred cartridges manufactured by each of the four companies were purchased by the FBI and the concentration of the five trace elements in the lead alloy tip was measured using the method of inductively coupled plasma spectrometry in the FBI laboratories ([35]). In the original study, the

model-based clustering algorithm proposed by Raftery ([4]) was implemented on the log-transformed 5-dimensional observation vectors separately for each manufacturer. After log-transformation, the observation vectors were assumed to be distributed as a multivariate normal mixture. While a different number of compositional groups was estimated for bullets from each of the four manufacturers, results suggested that in all cases, the best fitting model (using an information-based criterion such as Akaike's information criterion) was one where the component covariance matrices were not homogeneous. Therefore, carrying out tests for differences in the mean concentration vectors of each of the compositional groups required additional assumptions.

We use these data in Chapter 5 of this dissertation to illustrate the implementation of the methods we propose for modeling special orthogonal matrices and estimating the parameters in those models.

1.3 Thesis organization

This thesis is organized as follows. In Chapter 2 we define the Stiefel manifold, the appropriate space for orthonormal matrices, and analyze some of its properties. Chapter 3 includes a discussion of the matrix Langevin distribution defined on the Stiefel manifold and a review of classical methods to estimate the parameters of this particular distribution. We introduce a Bayesian approach to estimate the parameters of the matrix Langevin distribution in Chapter 4 and illustrate the methods in Chapter 5 by implementing them on one data analysis example. In Chapter 4 we also discuss in some detail the numerical methods required to implement the Bayesian approach to

estimation. Finally, the last chapter discusses some of the assumptions and extensions of this work. Appendices include data tables, computer code and additional figures and tables with results.

2 The Stiefel manifold $V_{m,n}$

2.1 Overview

In this chapter we lay the foundations required to describe probability models for special orthogonal matrices in Chapter 3. We first define the Stiefel manifold, the space of orthonormal matrices and list several special cases that have been discussed in the statistics literature. We then characterize the special orthogonal group $SO(n)$ of $n \times n$ rotation matrices and operations on the group. We focus on the characterization of the group as the group of rotation matrices in $\mathbb{R}^n$, and show how the rotation group can be written in terms of rotation angles. In the second half of the chapter, we first describe the Haar measure, an invariant measure on compact Riemannian manifolds, which permits calculation of the volume element and the total volume. We then focus on the Haar measure on the Stiefel manifold $V_{m,n}$ and on calculating the volume element on $V_{m,n}$. We show that the normalized Haar measure on $V_{m,n}$ is a probability measure and plays the same role on $V_{m,n}$ that the Lebesgue measure plays on $\mathbb{R}^n$.

2.2 Definition of Stiefel manifolds and the special orthogonal group

We denote by $M(n, m, \mathbb{R})$ the vector space of all $n \times m$ matrices with real entries. Let H be a $n \times m$, $n \geq m$ matrix such that $H'H = I_m$ where I_m is the $m \times m$ identity matrix. That is, the m columns of H are orthogonal vectors of Euclidian norm 1. The space of all such matrices H is known as the Stiefel manifold $V_{m,n}$. Thus,

$$V_{m,n} = \{H \in M(n, m, \mathbb{R}),\ H'H = I_m\}.$$

The points H in $V_{m,n}$ are called $m-$frames in $\mathbb{R}^n$ and $V_{m,n}$ defines a surface that is a subset of the sphere of radius $m^{1/2}$ in $M(n, m, \mathbb{R})$ with the Euclidian distance. This is a direct consequence of the fact that for $H = (h_{ij}) \in V_{m,n}$, $(i = 1, \ldots, n;\ j = 1, \ldots, m)$ we have that $\sum_{i=1}^{n} \sum_{j=1}^{m} h_{ij}^2 = m$. Furthermore, $V_{m,n}$ is determined by $m(m+1)/2$ functionally independent conditions on the mn elements of $H \in M(n, m)$. Note that these conditions are polynomial and hence, $V_{m,n}$ is an analytic manifold of dimension $mn - m(m+1)/2$.

Special cases of the Stiefel manifold are listed below.

1. For $m = 1$ we have

$$V_{1,n} = \{h \in M(n, 1, \mathbb{R}),\ h'h = 1\}.$$

Hence $V_{1,n}$ can be identified with the unit sphere $\mathbb{S}^{n-1} \subset \mathbb{R}^n$. The cases when $n = 2$ and $n = 3$ have been extensively studied in directional statistics ([42], [12], [13]).

2. For $m = n$ we have

$$V_{n,n} = \{H \in M(n, n, \mathbb{R}),\ H'H = I_n\},$$

which can be identified with the set $O(n)$ of $n \times n$ orthogonal matrices with columns of norm 1. This is a group, called the orthogonal group, with the group operation being matrix multiplication. Note that $O(n)$ can be regarded as a $n(n-1)/2-$dimensional surface in the n^2-dimensional Euclidean space $M(n, n, \mathbb{R}) \cong \mathbb{R}^{nn}$, and the surface is a subset of the sphere of radius $n^{1/2}$ in this space.

3. Another special case is when $m = n - 1$ which is the case of most interest to us. When $m = n - 1$, an orthogonal $(n-1)$-frame can be extended uniquely to an orthonormal $n-$frame with determinant 1, hence $V_{n-1,n}$ can be identified with $SO(n)$, the special orthogonal group consisting of all $n \times n$ rotation matrices. Next we will briefly construct this space for future reference.

Definition 1 *Let X be a set and $(G, \circ)$ a (not necessarily commutative) group with operation $\circ$ and unit $e \in G$. G is said to act on X (from the left) if there is a map $\bullet : G \times X \to X$ such that for all $x \in X$ and all $g_1, g_2 \in G$ the map satisfies two conditions: (i) $\bullet(e, x) = x$, and (ii) $\bullet(g_1 \circ g_2, x) = \bullet(g_1, \bullet(g_2, x))$. We often write $g \bullet x$ or even gx for $\bullet(g, x)$ if no confusion can arise. In this notation the conditions read (i)' $ex = x$ and (ii)' $(g_1 \circ g_2)x = g_1(g_2 x)$.*

Examples of group actions include, e.g.,

1. Invertible linear maps acting on a real vector space: $\bullet : Gl(n, \mathbb{R}) \times \mathbb{R}^n \to \mathbb{R}^n$, $\bullet(A, x) = Ax$, where $Gl(n, \mathbb{R})$ is the group of invertible $n \times n$ matrices.

2. Solutions of linear differential equations $\dot{x} = Ax$ with $A \in gl(n, \mathbb{R}) := M(n, n, \mathbb{R})$: Denote the solution of $\dot{x} = Ax$ with initial value $x \in \mathbb{R}^n$ at time $t = 0$ by $\varphi(t, x) = e^{At}x$. Then the set $\Phi = \{\varphi(t, \cdot), t \in \mathbb{R}\}$ is a group, with composition of maps as the group operation. The group Φ acts on $\mathbb{R}^n$ via $\bullet : \Phi \times \mathbb{R}^n \to \mathbb{R}^n$, $\bullet(e^{At}, x) = e^{At}x$. In a similar way, solutions of (nonlinear) differential equations $\dot{x} = f(x)$ in $\mathbb{R}^n$ can be considered as group actions, if the solutions are unique and exist for all times $t \in \mathbb{R}$.

3. Example 1 can be extended to other subgroups of $Gl(n, \mathbb{R})$, such as the orthogonal group $O(n)$, the special orthogonal group $SO(n)$, etc.

4. Example 3 can be extended from group actions on the space $\mathbb{R}^n$ to actions on subspaces, spaces of frames $V_{m,n}$, and Grassmannian manifolds of $\mathbb{R}^n$, as long as the group maps these spaces into themselves. An example is the action of the special orthogonal group $SO(n)$ on the sphere $\mathbb{S}^{n-1}$, since $Ax \in \mathbb{S}^{n-1}$ for $A \in SO(n)$ and $x \in \mathbb{S}^{n-1}$.

Definition 2 *Two points x_1, x_2 in X are said to be equivalent under G, written $x_1 \sim x_2$ mod(G), if there exists $g \in G$ such that $x_2 = gx_1$.*

Definition 3 *A function ϕ defined on X is said to be invariant under G if*

$$\phi(gx) = \phi(x) \text{ for all } x \in X \text{ and all } g \in G.$$

Definition 4 *If $x_1 \sim x_2$ mod(G) for all x_1, x_2 in X, then the group G is said to act transitively on X, and X is said to be homogeneous with respect to G.*

Note that if G acts transitively on X, then $G-$invariant functions on X are constant.

Definition 5 *For $x_0 \in X$ we define the subgroup G_0 of G, consisting of all transformations which leave x_0 invariant, namely*

$$G_0 = \{g \in G,\ gx_0 = x_0\},$$

as the isotropy group of G at x_0.

Definition 6 *Let G_0 be the isotropy group of G at $x_0 \in X$. For each $g \in G$ the set*

$$gG_0 = \{gg_0,\ g_0 \in G_0\} \subset G$$

is called a coset of G_0 in G. We define the quotient $G/G_0 := \{gG_0,\ g \in G\}$ as the set of cosets of G_0 in G.

Lemma 1 *(i) Assume that the group G acts transitively on X, and let G_0 be the isotropy group of G at some $x_0 \in X$. Then the map $\iota : G/G_0 \to X$ defined by $\iota(gG_0) = gx_0$ is bijective. Hence the set X can be considered as the quotient space G/G_0.*

(ii) Assume furthermore that the group G is a compact topological group, X is a compact Hausdorff space, and $\bullet$ is continuous. Then G_0 is closed and ι is a homeomorphism.

For a proof of Lemma 1 see Theorem II.3.2. in [22], and Sections III.6 and III.7 in [6].

We are now ready to characterize the special orthogonal group following the approach described on pp. 120-132 in [41]:

We take $X = V_{m,n}$ and $G = O(n)$. The action of $O(n)$ on $V_{m,n}$ is given by

$$\bullet : O(n) \times V_{m,n} \to V_{m,n}, \ \bullet (H, Q) = HQ$$

with the group operation being matrix multiplication. Note that $O(n)$ acts transitively on $V_{m,n}$. The isotropy subgroup of $O(n)$ at

$$\begin{bmatrix} I_m \\ \cdots \\ 0 \end{bmatrix} \in V_{m,n}$$

is

$$G_0 = \left\{ \begin{bmatrix} I_m & 0 \\ 0 & H_1 \end{bmatrix} \in O(n),\ H_1 \in O(n-m) \right\},$$

and the coset corresponding to $Q_1 \in V_{m,n}$ is $[Q_1, Q_2]G_0$ where Q_2 is any $n \times (n-m)$ such that $[Q_1, Q_2] \in O(n)$. This coset consists of all orthogonal $n \times n$ matrices with Q_1 as the first m columns. Writing the homogeneous space $V_{m,n}$ as the coset space of the isotropy group we have

$$V_{m,n} = O(n)/O(n-m).$$

Thus, for $m = n - 1$ we obtain

$$V_{n-1,n} \cong O(n)/O(1) = \{H \in O(n);\ \det H = 1\} \equiv SO(n).$$

Note that $SO(n)$ is a compact, connected, $n(n-1)/2$ dimensional Lie group, namely the connected component of $O(n)$ that contains the identity matrix I_n. We now have three characterizations of the special orthogonal group $SO(n)$:

- as the Stiefel manifold $V_{n-1,n}$,
- as the homogeneous space $O(n)/O(1)$, and
- as the group of rotation matrices in $\mathbb{R}^n$.

Each of these characterizations will be useful in the future.

Specifically, the rotation group can be described by $n(n-1)/2$ angles, generalizing the Euler angles to dimension n (e.g., [24]). To do so, we fix an (ordered) orthonormal basis $B = (b_1, ..., b_n)$ in $\mathbb{R}^n$ and denote by $R_{p,q}$, $p < q$ a rotation in the 2−dimensional subspace $ls(b_p, b_q) \subset \mathbb{R}^n$ spanned by (b_p, b_q). In terms of the rotation angle α_{pq} the matrix $R_{p,q}$ looks like

$$R_{p,q} = \begin{bmatrix} I_1 & 0 & 0 & 0 & 0 \\ 0 & \cos\alpha_{pq} & 0 & -\sin\alpha_{pq} & 0 \\ 0 & 0 & I_2 & 0 & 0 \\ 0 & \sin\alpha_{pq} & 0 & \cos\alpha_{pq} & 0 \\ 0 & 0 & 0 & 0 & I_3 \end{bmatrix} \begin{matrix} \\ \leftarrow \quad p-\text{th row} \\ \\ \leftarrow \quad q-\text{th row} \\ \\ \\ \uparrow \qquad \uparrow \\ p-\text{th column} \quad q-\text{th column} \end{matrix},$$

where the identity matrices I_1, I_2, and I_3 are of dimension $(p-1, p-1)$, $(q-p-1, q-p-1)$, and $(n-q, n-q)$, respectively. Any rotation matrix $X \in O(n)$ can then be written uniquely as a product

$$X = R_{n-1,n} \cdot \ldots \cdot R_{2,n} \cdot \ldots \cdot R_{2,3} \cdot R_{1,n} \cdot \ldots \cdot R_{1,3} \cdot R_{1,2}, \tag{2.1}$$

with $\alpha_{pq} \in [0, 2\pi)$. For $\mathbb{R}^3$ we obtain the classical Euler angles, described by the matrices

$$R_{1,2} = \begin{bmatrix} \cos\alpha_{12} & -\sin\alpha_{12} & 0 \\ \sin\alpha_{12} & \cos\alpha_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{1,3} = \begin{bmatrix} \cos\alpha_{13} & 0 & -\sin\alpha_{13} \\ 0 & 1 & 0 \\ \sin\alpha_{13} & 0 & \cos\alpha_{13} \end{bmatrix}$$

$$R_{2,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_{23} & -\sin\alpha_{23} \\ 0 & \sin\alpha_{23} & \cos\alpha_{23} \end{bmatrix}$$

as in the standard pitch-roll-yaw model ([2]). We argue later in Chapter 4 that representing rotation matrices using Euler angles greatly increases the efficiency of algorithms for drawing rotation matrices from their posterior distribution.

2.3 Volume and Haar Measure on $V_{m,n}$

In this section we use the characterization of the Stiefel manifold $V_{m,n}$ as a Riemannian manifold (see, e.g., [6] for details on differential forms and exterior products). Let Z be a $n \times m$ $(n \geq m)$ matrix of rank m. We decompose Z into its orthogonal and triangular factor as $Z = H_1 T$, where H_1 is an $n \times m$ matrix with $H_1' H_1 = I_m$ and $T = (t_{ij})$, $i, j = 1, ...m$ is a $m \times m$ upper-triangular matrix with positive diagonal elements. Let H_2 (a function of H_1) be an $n \times (n - m)$ matrix such that $H = [H_1 : H_2] \in O(n)$. We write

$H = [h_1, \ldots, h_m : h_{m+1}, \ldots, h_n]$ where $h_1, \ldots, h_m$ and $h_{m+1}, \ldots, h_n$ are the columns of H_1 and H_2, respectively. Then the differential dZ is defined as

$$(dZ) = \prod_{i=1}^{m} t_{ii}^{n-i}(dT)(H_1' dH_1),$$

where

$$(H_1' dH_1) = \bigwedge_{i=1}^{m} \bigwedge_{j=i+1}^{n} h_j' dh_i,$$

and the symbol $\bigwedge$ denotes the exterior product.

2.3.1 The orthogonal group $O(n)$

We first consider the special case when $n = m$. That is, for $H \in O(m)$

$$H'dH = \bigwedge_{i<j}^{m} h_j' dh_i.$$

This differential form is the exterior product of the subdiagonal elements of the skew-symmetric matrix $H'dH$ and it has the following properties:

1. It is invariant under *left* translations:

   Let $\Phi_Q : O(n) \to O(n)$ be defined by $H \to QH$ for $Q \in O(n)$, then

   $$H'dH \to H'Q'dQH = H'Q'QdH = H'dH.$$

2. It is invariant under *right* translations:

   Let $\Phi^Q : O(n) \to O(n)$ be defined by $H \to HQ$ for $Q \in O(n)$, then

   $$H'dH \to Q'H'dHQ = (\det Q)^{n-1} H'dH = H'dH$$

(ignoring the sign), since if dy is an $m \times 1$ vector of differentials and if $dx = Bdy$, where B is an $m \times m$ nonsingular matrix, then

$$\bigwedge_{i=1}^{m} dx_i = \det B \bigwedge_{i=1}^{m} dy_i.$$

The differential form $H'dH$ defines a measure μ on the Borel $\sigma-$algebra $\mathfrak{B}(O(n))$ given by

$$\mu(D) = \int_D (H'dH) \text{ for } D \in \mathfrak{B}(O(n)),$$

with $\mu(D)$ interpreted as the area of the region D in the $O(n)$.

The measure μ is (left and right) invariant, i.e.

$$\mu(QD) = \mu(DQ) = \mu(D) \text{ for all } Q \in O(n).$$

This measure μ is called the invariant measure or the Haar measure on $O(n)$. The measure μ is unique in the sense that any other translation invariant measure on $O(n)$ is a finite multiple of μ.

By means of this measure the total volume of $O(n)$ can be calculated as

$$\text{Vol}[O(n)] = \mu(O(n)) = \int_{O(n)} (H'dH).$$

2.3.2 The Stiefel manifold $V_{m,n}$

To define the Haar measure and the volume on $V_{m,n}$ we follow a different route that is more appropriate for statistical computations. We first need to introduce the multivariate gamma function. The multivariate gamma function of dimension m and argument $a > (m-1)/2$ is defined as

$$\Gamma_m(a) = \int_{A>0} \text{etr}(-A)(\det A)^{a-(m+1)/2} dA,$$

where $\text{etr}(-A) = \exp(trA)$, and the integral is over all positive definite matrices A. It can be shown that $\Gamma_m(a) = \pi^{m(m-1)/4} \prod_{i=1}^{m} \Gamma(a - (i-1)/2)$, where Γ is the standard $1-$dimensional gamma function ([26]).

Now, let Z be an $n \times m$ random matrix whose elements are independent $N(0,1)$ random variables. We set

$$f(Z) = (2\pi)^{mn/2} \text{etr}(-\frac{1}{2} Z'Z),$$

and note that

$$\int_{\mathbb{R}^{mn}} \text{etr}(-\frac{1}{2} Z'Z)(dZ) = (2\pi)^{-mn/2}. \tag{2.2}$$

As above, we decompose $Z = H_1 T$ with $H_1 \in V_{m,n}$ and Z upper-triangular with positive diagonal elements. Then,

$$\text{tr}(Z'Z) = \text{tr}(T'H_1'H_1T) = \text{tr}(T'T) = \sum_{i \le j}^{m} t_{ij}^2$$

and

$$(dZ) = \prod_{i=1}^{m} t_{ii}^{n-i} dT (H_1' dH_1).$$

Thus the left hand side of (2.2) is

$$\int_{\mathbb{R}^{mn}} \exp\left[-\frac{1}{2} \sum_{i \le j}^{m} t_{ij}^2\right] \prod_{i=1}^{m} t_{ii}^{n-i} \bigwedge_{i \le j}^{m} dt_{ij} \int_{V_{m,n}} (H_1' dH_1).$$

The integral involving the t_{ij} can be written as

$$\begin{aligned}
&\prod_{i\le j}^{m}\int_{-\infty}^{\infty}\exp(-t_{ij}^2/2)dt_{ij}\times\prod_{i=1}^{m}\int_{0}^{\infty}\exp(-t_{ii}^2/2)t_{ii}^{n-i}dt_{ii}\\
&=\left[\prod_{i\le j}^{m}(2\pi)^{\frac{1}{2}}\right]\left[\prod_{i=1}^{m}2^{(n-i-1)/2}\Gamma[(n-i+1)/2]\right]\\
&=\pi^{m(m-1)/4}\left[\prod_{i=1}^{m}\Gamma[(n-i+1)/2]\right]2^{mn/2-m}\\
&=\Gamma_m(n/2)\times 2^{mn/2-m}.
\end{aligned}$$

We then obtain

$$\mathrm{Vol}[V_{m,n}]=\int_{H_1\in V_{m,n}}(H_1'dH_1)=\frac{2^m\pi^{mn/2}}{\Gamma_m(n/2)}.$$

The measure $H_1'dH_1$ defined above on $V_{m,n}$ is an "unnormalized" measure. It can be normalized to a probability measure by setting

$$[dH]=\frac{1}{\mathrm{Vol}[V_{m,n}]}(H_1'dH_1)=\frac{\Gamma_m(n/2)}{2^m\pi^{mn/2}}\bigwedge_{i\le j}^{m}h_j'dh_i,$$

i.e.

$$\int_{V_{m,n}}(dH)=1.$$

We denote this normalized measure by μ^*, i.e.

$$\mu^*(D)=\int_D(dH)\text{ for }D\in\mathfrak{B}(V_{m,n})$$

is a probability measure.

The measure $\mu^*(\cdot)$ has the following properties

1. $\mu^*(\cdot)$ is left invariant under the action of $O(n)$ on $V_{m,n}$, i.e.

$$\mu^*(QD)=\mu^*(D)\text{ for all }Q\in O(n).$$

2. $\mu^*(\cdot)$ is right invariant under the action of $O(m)$ on $V_{m,n}$, i.e.

$$\mu^*(DQ) = \mu^*(D) \text{ for all } Q \in O(m).$$

The measure μ^* is called the Haar measure on $V_{m,n}$. Because it is translation invariant, it plays the same role on $V_{m,n}$ that the Lebesgue measure plays on $\mathbb{R}^n$, but since $V_{m,n}$ is compact, this measure is finite. It is therefore the uniform distribution on $V_{m,n}$. Indeed, $\mu^*(\cdot)$ is the unique probability measure on $V_{m,n}$ which is invariant under rotations and reflections, i.e. if X is a random matrix whose distribution is uniform on $V_{m,n}$ then, UXV has the same distribution as X for all $U \in O(n)$ and $V \in O(m)$.

As an example on how to calculate measures on $V_{m,n}$, consider $SO(2)$. If $H \in SO(2)$, H can be parameterized as

$$H = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = [h_1 \; h_2],\ 0 \leq \theta < 2\pi,$$

as was described toward the end of Section 2.2 when we characterized $SO(2)$ in terms of angles. From this parametrization we compute the volume of $SO(2)$ as

$$H'dH = h_2'dh_1 = (-\sin\theta \;\; \cos\theta) \begin{bmatrix} -\sin\theta d\theta \\ \cos\theta d\theta \end{bmatrix} = d\theta,$$

and hence

$$\text{Vol}[SO(2)] = \int_{SO(2)} H'dH = \int_0^{2\pi} d\theta = 2\pi.$$

3 The matrix Langevin distribution on $V_{m,n}$

3.1 Overview

In this chapter we describe the matrix Langevin distribution on the Stiefel manifold and illustrate some of its properties using the special case of the one-dimensional Langevin distribution also known as the von Mises distribution. The Langevin distribution is parametrized by a matrix of parameters F that can be estimated from a classical or a Bayesian framework. In this chapter, we implement the maximum likelihood approach to estimate F given a sample of orthogonal matrices, and delay discussion of the Bayesian approach to estimation until the next chapter.

We carry out maximum likelihood estimation by first decomposing F into three component matrices using its singular value decomposition, and write the likelihood function in terms of the singular value components. Under some assumptions, we derive MLEs of the elements of F and discuss some of their properties.

Results presented in this chapter draw from earlier work in [9] and others. Here, we extend some of the results in [9] in our Section 3.4.

3.2 Definition of the matrix Langevin distribution

Let $V_{m,n}$ be the Stiefel manifold of $m-$frames in $\mathbb{R}^n$ and μ^* the normalized Haar measure on $V_{m,n}$ as described in Section 2.3. As noted above, the probability measure μ^* is the (unique) probability distribution on $V_{m,n}$ that is invariant under the (left) action of $O(n)$, describing the uniform distribution on $V_{m,n}$. Hence it serves as the reference measure on $V_{m,n}$ just as the Lebesgue measure serves as the reference measure on $\mathbb{R}^n$. Given a sample of observations, it is possible to test whether the data could have been plausibly generated by this distribution by means of the Rayleigh test as described, for example, in [30].

Based on the normalized Haar measure μ^*, a family of probability distributions has been defined by [11] on $V_{m,n}$ in the following way. Let F be a $n \times m$ matrix of parameters, denote the volume element of μ^* by $[dX]$, and define

$$dF(X) = \begin{cases} a(F)\exp\{tr(F'X)\}[dX] & X \in V_{m,n} \\ 0 & \text{otherwise,} \end{cases}$$

where $a(F)$ is a normalizing constant and $tr(A)$ denotes the trace of a matrix A. This family of distributions is called the matrix (variate) Langevin distribution. Special cases are the well known univariate *von Mises* distribution used to model distribution of points on a circle, and the bivariate *Fisher-von Mises* distribution used to model distribution of points on a sphere ([12]; [13]; [31]).

Just for purposes of illustration we focus briefly on the *von Mises* distribution ([13]). The distribution is the circular analog of the normal distribution on a line and is indexed by two parameters α, b which control the mean direction and the concentration,

respectively. For a scalar-valued random variable $x \in [0, 2\pi)$, the density function is given by

$$f(x|\mu, \kappa) = \frac{\exp[\kappa \cos(x - \mu)]}{2\pi I_0(\kappa)},$$

where $I_0(\kappa)$ is a modified Bessel function of the first kind of order 0. Figure 3.1 below shows the *von Mises* distribution on the circle for a range of values of κ for $\mu = 0$.

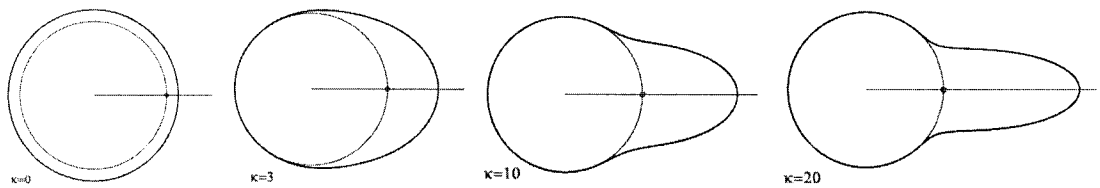

Figure 3.1 Plot of the von Mises distribution on the circle for $\mu = 0$ and selected values of κ.

Khatri and Mardia ([31]) have shown that the normalizing constant $\frac{1}{a(F)}$ of the matrix Langevin distribution can be written as a function of zonal polynomials, or alternatively as the hypergeometric function ${}_0F_1(\frac{1}{2}m, \frac{1}{4}F'F)$ (with matrix argument F), ([26]; [38]). Thus, we will say that a random matrix X on $V_{m,n}$ has the matrix Langevin distribution, if its density function is given by

$$\frac{1}{{}_0F_1(\frac{1}{2}m; \frac{1}{4}F'F)} \text{etr}(F'X),$$

where, as before, $\text{etr}(A) = \exp(tr(A))$ for a matrix A, and

$${}_0F_1(\frac{1}{2}m; \frac{1}{4}F'F) = \int_{V_{m,n}} \text{etr}(F'H)[dH].$$

To derive methods for estimating the matrix of parameters F, we use two different representations, although one can be seen as a slight variation of the other. The first representation is obtained by decomposing the matrix F into the product of two matrices,

i.e. $F = MK$ where $M \in V_{m,n}$ and K is a $m \times m$ symmetric positive semi-definite matrix. The matrix M is called the polar part or orientation, and K is the elliptical part or concentration of F ([30]). The components M and K are analogous to the parameters α and b in the one-dimensional von Mises distribution described earlier. A second approach, which serves better for our purposes, decomposes a (full rank) matrix F into the product of three matrices as follows. Let

$$F = \Gamma D_\phi \Delta$$

be the singular value decomposition of F, where D_ϕ =diag$(\phi_1, \ldots, \phi_m)$, $\Gamma \in \widetilde{V}_{m,n}$, and $\Delta \in O(m)$, with the following properties:

1. We assume that the singular values satisfy $\phi_1 > \phi_2 > \ldots > \phi_m$ for F of full rank.

2. We denote by $\widetilde{V}_{m,n}$ the space of matrices Γ in $V_{m,n}$ with the property that all the elements of the first row of the matrix (of left eigenvectors) Γ are positive. If some of those elements are zero, the first nonzero elements of those columns, whose first element is equal to zero, have to be positive ([21]).

3. No specific restrictions need to be imposed on Δ since the columns of Γ uniquely determine the columns of Δ.

These conditions guarantee uniqueness of the singular value decomposition $F = \Gamma D_\phi \Delta$. Then it can be shown ([31]) that ${}_0F_1(\frac{1}{2}m, \frac{1}{4}F'F) = {}_0F_1(\frac{1}{2}m, \frac{1}{4}D_\phi^2)$. That is, the normalizing $a(F)$ constant depends only on D_ϕ.

It is straightforward to see that this distribution is part of the family of exponential distributions ([8]).

3.3 Maximum Likelihood Estimation

Let $X_1, X_2, \ldots, X_d$ be an i.i.d. sample from the matrix Langevin distribution with parameter F on the Stiefel manifold $V_{m,n}$, and set $\overline{X} = \frac{1}{d}\sum X_i$. The log-likelihood function for this sample is

$$l(X_1, \ldots, X_d|F) = d\left[tr(\overline{X}'F) - \log\ {}_0F_1(\frac{1}{2}m; \frac{1}{4}F'F)\right].$$

Thus, $\overline{X}$ is sufficient for F. However, $\overline{X}$ may not be a suitable estimator for F since it may not be even a point in the space $V_{m,n}$. Consider, for example, the Stiefel manifold $SO(2)$ and A_1, $A_2 \in SO(2)$. In the parametrization as a submanifold of $M(2,2,R)$, $\frac{1}{2}(A_1 + A_2) \notin SO(2)$, unless $A_1 = A_2$.

Therefore, we rewrite the log-likelihood function using the singular value decomposition from Section 3.2. Let $\overline{X} = H_1 D_x H_2$ be the unique singular value decomposition of $\overline{X}$ with $H_1 \in \widetilde{V}_{m,n}$, $H_2 \in O(m)$, and D_x =diag$(x_1, \ldots, x_m)$ with $x_1 > \ldots > x_m$ almost everywhere. Thus, the log-likelihood function becomes

$$l(X_1, \ldots, X_d|F) = d\ tr(\Delta D_\phi \Gamma' H_1 D_x H_2) - d \log\ {}_0F_1(\frac{1}{2}m; \frac{1}{4}D_\phi^2).$$

Then, the maximum likelihood estimators $\hat{\Gamma}$, $\hat{\Delta}$, and $\hat{D}_\phi$ =diag$(\hat{\phi}_1, \ldots, \hat{\phi}_m)$ of Γ, Δ, and D_ϕ are given by H_1, H_2, and the system of differential equations

$$\frac{\partial \log\ {}_0F_1(\frac{1}{2}m; \frac{1}{4}\hat{D}_\phi^2)}{\partial \hat{\phi}_i} = x_i \qquad \text{for } i = 1, \ldots, m, \tag{3.1}$$

respectively ([31]).

In general, the system of differential equations (3.1) has to be evaluated numerically. However, when D_ϕ is large (i.e. the $\phi_1, \ldots, \phi_m$ are large), the hypergeometric function

can be approximated by ([31],[34])

$$ {}_0F_1(\frac{1}{2}m, \frac{1}{4}\hat{D}_\phi^2) \approx \frac{\Gamma_d\left(\frac{1}{2}m\right)}{2^{m-1}\phi^{m^2/2}} \exp\left(\sum_{i=1}^{m}\hat{\phi}_i\right) \prod_{i=1;j=1;i<j}^{m}\left(\frac{2\pi}{\hat{\phi}_i+\hat{\phi}_j}\right)^2, $$

where $\Gamma_d(\cdot)$ denotes again the multivariate gamma function of dimension d. Hence we obtain in this case an approximation of (3.1) as

$$ \frac{\partial \log\ {}_0F_1(\frac{1}{2}m, \frac{1}{4}\hat{D}_\phi^2)}{\partial\hat{\phi}_i} \approx 1 - \frac{1}{2}\sum_{i=1;j=1;i<j}^{m}\frac{1}{\hat{\phi}_i+\hat{\phi}_j}. \tag{3.2} $$

Thus, we need to solve the algebraic equations

$$ 1 - \frac{1}{2}\sum_{i=1;j=1;i<j}^{m}\frac{1}{\hat{\phi}_i+\hat{\phi}_j} = x_k \qquad k = 1,\ldots,m $$

for $\hat{\phi}_1,\ldots,\hat{\phi}_m$.

Alternatively, when D_ϕ is small, we have the approximation ([31],[34])

$$ \hat{\phi}_i \approx dx_i \qquad \text{for } i = 1,\ldots,m $$

and therefore

$$ {}_0F_1(\frac{1}{2}m, \frac{1}{4}\hat{D}_\phi^2) \approx 1 + \frac{1}{2d}\sum_{i=1}^{m}\widehat{\phi}_i^2. $$

An approximate solution to (3.1) is then obtained by solving the system of differential equations

$$ \frac{\partial}{\partial\hat{\phi}_i}(1 + \frac{1}{2d}\sum_{i=1}^{m}\widehat{\phi}_i^2) = x_i \qquad \text{for } i = 1,\ldots,m. \tag{3.3} $$

It is important to note that the real magnitude of the ϕ_i's is unknown, thus, as a rule of thumb, the magnitude of the x_i's is used to determine whether the ϕ_i's are large or small.

3.4 Properties of the MLE

We assume in this section that the diagonal matrix D_ϕ in the decomposition $F = \Gamma D_\phi \Delta$ of the parameter matrix F is known, and we first assume that F is of full rank m. Then we can establish the following properties of the MLE estimator. The results below extend some of the results presented by [28] and [9].

1. If $\hat{\Gamma}(F)$ is the MLE estimator of Γ in $F = \Gamma D_\phi \Delta$, then

$$\hat{\Gamma}(HF) = H\hat{\Gamma}(F) \text{ for all } H \in O(n),$$

   i.e. the MLE of Γ is equivariant under left translation with $O(n)$.

2. If $\hat{\Delta}(F)$ is the MLE estimator of Δ in $F = \Gamma D_\phi \Delta$, then

$$\hat{\Delta}(FH) = \hat{\Delta}(F)H \text{ for all } H \in O(m).$$

3. For any loss function L satisfying

$$L(H\Gamma, H\hat{\Gamma}) = L(\Gamma, \hat{\Gamma}) \text{ for all } H \in O(n)$$

   $\hat{\Gamma}$ is the minimum risk equivariant estimator of Γ.

4. For any loss function L satisfying

$$L(\Gamma, \hat{\Gamma}; \Delta, \hat{\Delta}) = L(H_1\Gamma, H_1\hat{\Gamma}; \Delta H_2, \hat{\Delta}H_2) \text{ for all } H_1 \in O(n),\ H_2 \in O(m)$$

   $\hat{\Gamma}$ and $\hat{\Delta}$ are simultaneously invariant and $\hat{\Gamma}$ and $\hat{\Delta}$ are the minimum risk invariant estimators of Γ and Δ.

5. The MLE estimators $\hat{\Gamma}$ and $\widehat{\Delta}$ are consistent.

Remark 1 *In general, the MLE estimators* $\hat{\Gamma}$ *and* $\widehat{\Delta}$ *are not unbiased. Indeed, unbiasedness of MLE estimators is not necessarily a desirable property on manifolds such as* $V_{m,n}$*, as the following reasoning for* $V_{m,n} = O(n)$ *shows. If* $\hat{\Gamma}$ *is unbiased then* $\Gamma = E(\hat{\Gamma})$ *by definition, and together with the fact that* $\Gamma \in O(n)$ *this implies that*

$$\Gamma = E(\hat{\Gamma}) = E(\Gamma\Gamma'\hat{\Gamma}) = \Gamma E(\Gamma'\hat{\Gamma}).$$

Thus, for $\hat{\Gamma}$ *to be unbiased we need* $E(\Gamma'\hat{\Gamma}) = I_n$*. On the other hand we have for all* $A \in O(n)$

$$\mathcal{L}(\hat{\Gamma}) = \mathcal{L}(A\hat{\Gamma}),$$

where $\mathcal{L}$ *denotes the distribution of* $\hat{\Gamma}$*. This implies that*

$$E(\hat{\Gamma}) = E(A\hat{\Gamma}).$$

Take $A = \Gamma'$*, then*

$$E(\hat{\Gamma}) = E(\Gamma'\hat{\Gamma}) = I_n,$$

and hence $\Gamma = I_n$*.*

If the rank of F is not known a priori, we can proceed as follows to test its dimensionality ([31]). Let rank(F)=r where $0 \leq r \leq m$. We wish to test

$$H_0 : r = p \qquad \text{against} \qquad H_1 : r = m$$

sequentially for $p = 1, \ldots, m-1$. It can be shown that if λ is the test criterion for this problem, then a sampling distribution for $-2\log\lambda$ is given by

$$-2\log\lambda = 2d \left\{ \begin{array}{c} \log\ {}_0F_1(\frac{d}{2}, \frac{1}{4}\hat{D}_{\hat{\phi}^2}) - \log\ {}_0F_1(\frac{d}{2}, \frac{1}{4}\hat{D}_{\hat{\hat{\phi}}^2}) + \\ + \sum_{i=1}^{m} \hat{\hat{\phi}}_i g_i - \sum_{i=1}^{m} \hat{\phi}_i g_i \end{array} \right\}$$

$$\sim \chi^2_{(m-r)d},$$

where

1. d is the sample size,

2. $g_1^2, \ldots, g_m^2$ $(g_1^2 > \ldots > g_m^2)$ are the eigenvalues of $\bar{X}'\bar{X}$,

3. $\hat{\phi}^2$ are the MLE's of ϕ_i $(i = 1, \ldots, m)$ under H_0,

4. $\hat{\hat{\phi}}^2$ are the MLE's of ϕ_i $(i = 1, \ldots, m)$ under H_1.

Therefore, a test of the hypothesis $H_0 : r = p$ against $H_1 : r = m$ at level α would reject the null when $-2\log\lambda \geq \chi^2_{(m-r)d}(\alpha)$, where $\chi^2_{(m-r)d}(\alpha)$ is the upper $100(1-\alpha)$ percentile of a χ^2 distribution with $(m-r)d$ degrees of freedom.

4 Bayesian estimation in the matrix Langevin distribution

4.1 Overview

Although today's computer capabilities have made Bayesian methods relatively easy to apply even in complex models, few attempts have been made to apply them to estimation and inference on (higher dimensional) Stiefel manifolds $V_{m,n}$. The literature shows some applications to low-dimensional matrix Langevin distributions in the context of directional statistics, i.e. to the study of distributions on the circle in $\mathbb{R}^2$ or the sphere in $\mathbb{R}^3$. Examples include [33], [3], and [32]. Note that many of these results pre-date the introduction of Markov chain Monte Carlo methods as a tool for approximating posterior distributions, which only occurred in the early 1990s ([17]). Therefore, most of what has been proposed in terms of Bayesian estimation methods on probability models for orthogonal matrices has relied on analytical derivations. This has limited the scope and complexity of the applications that can be addressed with those methods.

We describe a set-up on $V_{m,n}$ for a general m and n, with specific attention to the orthogonal group $O(n) \cong V_{n,n}$ and the special orthogonal group $SO(n) \cong V_{n-1,n}$ that appear naturally in the applications of the next chapter. We rely on Markov chain Monte Carlo methods to approximate posterior distributions of parameters of interest

and therefore also describe in some detail efficient algorithms that can be used to generate orthogonal matrices from the appropriate distributions. Chikuse ([9]) and Hoff ([23]) have proposed approaches for generating uniformly-distributed orthogonal matrices and we adopt and extend some of these results in Section 4.4.1.

The application of Markov chain Monte Carlo methods requires sampling from the normalized Haar measure and the matrix Langevin distribution. Given $X_1, \ldots, X_d$, an i.i.d. sample of size d from a matrix Langevin distribution with parameter F, one can construct a Markov chain with $p(F|X_1, \ldots, X_d)$ as its stationary distribution via Gibbs sampling ([37]). Further, by considering as before the singular value decomposition (SVD) of F, i.e $F = \Gamma D_\phi \Delta$, $p(\Gamma, D_\phi, \Delta|X_1, \ldots, X_d)$ can be approximated by iteratively sampling from their full conditional distributions.

The SVD of F consists of two parts that are compact (Γ and Δ) and a part that is unbounded (D_ϕ). We will assume first that D_ϕ is known and later we discuss why relaxing this assumption is rather challenging. As it was noted before, the normalizing constant of the matrix Langevin distribution depends only on D_ϕ. Hence, by assuming D_ϕ to be known, the constant $\frac{1}{a(F)}$ can be computed beforehand using the hypergeometric function ${}_0F_1(\frac{1}{2}m, \frac{1}{4}D_\phi^2)$ since there is no need to update the value of $\frac{1}{a(F)}$ at each iteration of the Gibbs sampler.

4.2 Distribution of the sample sum on $V_{m,n}$

The matrix Langevin distributions form an exponential family. The likelihood can be written as

$$p(S_d|F) \propto \prod_{i=1}^{d} \text{etr}(F'X_i) = \text{etr}(F'S_d),$$

where

$$S_d = \sum_{i=1}^{d} X_i,$$

and $X_i \in V_{m,n}$. By using the SVD of $F = \Gamma D_\phi \Delta$ and assuming that D_ϕ is a fixed known constant matrix we can then write

$$p(S_d|\Gamma, \Delta) \propto \text{etr}\{(\Gamma D_\phi \Delta)' S_d\} = \text{etr}(\Delta' D_\phi \Gamma' S_d).$$

Note that the distribution of the sample sum is not a matrix Langevin distribution as the sample sum does not belong to $V_{m,n}$. The exact form of the distribution of the sample sum is ([9]):

$$p(S_d|\Gamma, \Delta) = \frac{1}{[{}_0F_1(m/2, D_\phi^2/4)]^d} \text{etr}(F'S_d) f_{S_d}(S_d; 0),$$

where

$$f_{S_d}(S_d; 0) = \frac{1}{(2\sqrt{\pi})^{nm}\Gamma_m(n/2)} \int_{U>0} {}_0F_1(n/2, -S_d U S_d') \times [{}_0F_1(n/2, -U/4)]^d |U|^{(n-m-1)/2}(dU).$$

The above integral is taken over the space of all $m \times m$ positive definite matrices.

4.3 Choice of prior distributions and resulting full conditional distributions

We complete the specification of the probability model for Γ and Δ by choosing prior distributions for the matrices. The joint posterior distribution of Γ and Δ is proportional to the product of the prior and the likelihood function described in Section 4.2.

Implementation of the Gibbs sampler to approximate the posterior distribution $p(\Gamma, \Delta | S_d, D_\phi)$ requires generating values of Γ, Δ from the full conditional distributions

$$p(\Gamma | S_d, D_\phi, \Delta), \quad p(\Delta | S_d, D_\phi, \Gamma).$$

Therefore, we derive the full conditional distributions for each choice of priors.

We assume that a priori, and conditional on D_ϕ, the matrices Γ and Δ are independent, even though this is clearly not a realistic assumption even after conditioning on D_ϕ (see Section 3.2 and the additional discussion in the next section). Under the assumption of prior independence, the joint prior density can be written as the product of two prior densities as follows

$$p(\Gamma, \Delta | D_\phi) = p(\Gamma | D_\phi) \times p(\Delta | D_\phi).$$

Two possible choices for prior distributions for Γ and Δ include

1. a non-informative prior distribution,

2. the conjugate prior distribution.

The non-informative prior distribution can be used to reflect our lack of prior knowledge about the parameter values. The conjugate distribution (e.g., [5]) can be more or

less informative depending on the value of its parameters, and can therefore be used to incorporate prior knowledge about Γ and Δ into the analysis. The conjugate family is mathematically convenient in that it leads to posterior distributions with the same form as the prior.

4.3.1 The lack of prior independence of Γ and Δ

For a given $n \times m$ matrix A of rank r $(0 < r \leq m \leq n)$, its singular value decomposition is of the form PDQ', where $P \in V_{m,n}$, $Q \in O(m)$ and D is a diagonal matrix with r diagonal elements that are strictly positive. Alternatively, let P_1 and Q_1 be the matrices that are obtained by taking the first r columns of P and Q respectively. The singular value decomposition of A can be rewritten as $P_1 D_1 Q_1'$ where D_1 is a diagonal matrix with the r nonzero elements of D on its diagonal ([21], [29]). Given A and D_1 the following relationships are true ([21])

$$\begin{aligned} P_1 &= AQ_1 D_1^{-1} \\ Q_1 &= A' P_1 D_1^{-1}. \end{aligned}$$

Hence, there is an algebraic dependence between P_1 and Q_1; that is, given P_1, Q_1 is known and vice versa. In our problem, the singular value decomposition of the matrix of parameters F is written as $\Gamma D_\phi \Delta$. Following the argument above, it is possible to write Δ as an algebraic function of Γ for a given F. Hence, the assumption of prior independence of Γ and Δ given D_ϕ is questionable. In the remainder, however, we do assume independence of the two matrices when building the joint prior distribution for (Γ, Δ), but propose that a future research objective would be to explicitly make use of

the algebraic dependence between the matrices to derive a more appropriate joint prior.

4.3.2 Non-informative prior distribution

The natural choice for a non-informative prior distribution is the Haar invariant measure μ^* on $V_{m,n}$ which plays the role of the uniform distribution on the Stiefel manifold, as was argued in Section 2.3. Choosing the prior distributions of $\Gamma \in \widetilde{V}_{m,n}$ and $\Delta \in O(m)$ independently, we can write

$$p(\Gamma, \Delta | D_\phi) = p(\Gamma | D_\phi) p(\Delta | D_\phi) \propto 1,$$

where p now denotes the respective Haar measure.

By choosing this distribution as our prior distribution we find that the joint posterior distribution of Γ and Δ is

$$p(\Gamma, \Delta | S_d, D_\phi) = \frac{1}{{}_0F_1^{(m)}(m/2, D_\phi^2, S_d' S_d)} \text{etr}(S_d' \Gamma D_\phi \Delta),$$

where ${}_0F_1^{(m)}(a, A, B)$ is the hypergeometric function of two matrix arguments ([34]).

Hence, the full conditional distribution of Γ is of the form

$$p(\Gamma | \Delta, S_d, D_\phi) \propto 1 \times \text{etr}(\Delta' D_\phi \Gamma' S_d) = \text{etr}(S_d \Delta' D_\phi \Gamma'),$$

and therefore the conditional distribution of Γ given Δ, D_ϕ and S_d is also a matrix Langevin distribution with parameter $F_\Gamma = D_\phi \Delta S_d'$. Similarly, we have

$$p(\Delta | \Gamma, S_d, D_\phi) \propto 1 \times \text{etr}(\Delta' D_\phi \Gamma' S_d) = \text{etr}(S_d' D_\phi \Gamma \Delta),$$

and thus the conditional distribution of Δ given Γ, D_ϕ and S_d is a matrix Langevin distribution with parameter $F_\Delta = \Gamma' D_\phi S_d$.

4.3.3 Conjugate prior distribution

In this section, we focus on the Stiefel manifold $V_{m,n}$ for $m = n$, i.e., on the orthogonal group $O(n)$, because in this case both Γ and Δ are elements of $O(n)$. Hence, the conjugate prior distribution for Γ is a matrix Langevin distribution with parameter A, for some $n \times n$ matrix A. Similarly, the conjugate prior distribution for Δ is a matrix Langevin distribution with parameter B, for some $n \times n$ matrix B. This means that

$$p(\Gamma, \Delta) = p(\Gamma)p(\Delta) \propto \text{etr}(A'\Gamma)\text{etr}(B'\Delta) = \text{etr}(A'\Gamma + B'\Delta),$$

and that the joint posterior distribution of Γ, Δ is therefore

$$p(\Gamma, \Delta | S_d, D_\phi) \propto \frac{1}{{}_0F_1^{(m)}(m/2, D_\phi^2, S_d'S_d)}\text{etr}(S_d'\Gamma D_\phi \Delta)\text{etr}(A'\Gamma + B'\Delta).$$

Hence the full conditional distribution of Γ with respect to this prior is

$$p(\Gamma | \Delta, S_d, D_\phi, A) \propto \text{etr}(A'\Gamma)\text{etr}(\Delta' D_\phi \Gamma' S_d) = \text{etr}\{(A' + S_d\Delta' D_\phi)\Gamma'\}.$$

In other words, the conditional distribution of Γ is a matrix Langevin distribution with parameter $F_\Gamma = (A' + S_d\Delta' D_\phi)'$. The conditional distribution of Δ given Γ and S_d is a matrix Langevin distribution with parameter $F_\Delta = (B' + S_d' D_\phi \Gamma)'$, which can be written as

$$p(\Delta | \Gamma, S_d, D_\phi, B) \propto \text{etr}(B'\Delta)\text{etr}(\Delta' D_\phi \Gamma' S_d) = \text{etr}\{(B' + S_d' D_\phi \Gamma)\Delta\}.$$

4.4 Generating random matrices on $V_{m,n}$

4.4.1 Samples from the Langevin distribution

A sample of pseudo-random matrices from the Langevin distribution can be obtained by using an acceptance-rejection method as described by Chikuse ([9]). To sample from the matrix Langevin distribution, with parameter $F = \Gamma D_\phi \Delta$, we generate a pseudo-random $n \times m$ matrix Q^* from the uniform distribution on $V_{m,n}$ and a pseudo-random variable u uniform on the interval $(0, 1)$. If $u <$etr$(F'X - D_\phi)$, we accept Q^* as a matrix in a sample with the desired density function. Otherwise, we repeat the procedure until a matrix Q^* is accepted.

4.4.2 Samples from the Haar invariant measure

There are several methods that can be used to generate pseudo-random matrices from the Haar invariant measure on $V_{m,n}$. Here, we discuss three approaches.

1. Generate nm independent pseudo-random $N(0, 1)$ variables. Arrange them in an $n \times m$ matrix Y, and let $Q^* = Y(Y'Y)^{-1/2}$. Then Q^* is the desired pseudo-random matrix from the uniform distribution on $V_{m,n}$ ([9]).

2. The second method is a variation of the approach proposed by Hoff ([23]). The procedure is as follows: For $X \in M(n, m, \mathbb{R})$ denote by X_i the $i-$th column of X and let N_i be any $n \times (n - i)$ matrix whose columns are an orthonormal basis of the null space of $X_{1\ldots i}$:$\mathbb{R}^n \to \mathbb{R}^m$, $X_{1\ldots i}(y) = (X_{1\ldots i})'y$.

- Generate n pseudo-random $N(0,1)$ variables and arrange them in a column vector y_1. Set $U_1 = y_1(y_1'y_1)^{-1/2}$.
- Calculate the null space N_1 of U_1 as above.
- For $j = 2, \ldots, m$:
  (a) Generate $n - j + 1$ pseudo-random $N(0,1)$ variables and arrange them in a vector y_j. Set $u_j = y_j(y_j'y_j)^{-1/2}$.
  (b) Set $U_j = (N_{j-1})u_j$.

The matrix $Q^* = (U_1, \ldots, U_m)$ is the desired pseudo-random matrix from the normalized Haar measure on $V_{m,n}$.

3. Focusing on $V_{m,n}$ as the group $SO(n)$ (i.e., $m = n - 1$), we can use the characterization of $SO(n)$ via angles that we described in Section 2.2 to generate samples from the normalized Haar measure on $SO(n)$. Recall that by (2.1) any rotation matrix $X \in SO(n)$ can then be written uniquely as a product $X = R_{n-1,n} \cdot \ldots \cdot R_{2,n} \cdot \ldots \cdot R_{2,3} \cdot R_{1,n} \cdot \ldots \cdot R_{1,3} \cdot R_{1,2}$ of angular rotations about $\alpha_{pq} \in [0, 2\pi)$ in the plane spanned by the vectors (b_p, b_q), $p = 1, \ldots, n-1$, $q = p+1, \ldots, n$ of an orthonormal basis $B = (b_1, \ldots, b_n)$ in $\mathbb{R}^n$. On the parametrization space $[0, 2\pi)^{n(n-1)/2}$ the normalized Haar measure μ^* is given by the product measure

$$\mu^* = \mu_{12} \otimes \mu_{13} \otimes \ldots \otimes \mu_{n-1,n},$$

where μ_{pq} is the uniform (probability) distribution on $[0, 2\pi)$ (with density $\frac{1}{2\pi} \cdot 1$): Obviously, each rotation $R_{p,q}$ leaves μ_{pq} invariant, and hence any rotation X leaves the product measure μ^* invariant.

Therefore, to sample a pseudo-random matrix from the Haar measure on $SO(n)$ we generate $n(n-1)/2$ pseudo-random variables θ_{pq} from the uniform distribution on $[0, 2\pi)$. For each (p, q) we obtain $R_{p,q}$. Then the matrix $Q^* = R_{n-1,n} \cdot \ldots \cdot R_{2,n} \cdot \ldots \cdot R_{2,3} \cdot R_{1,n} \cdot \ldots \cdot R_{1,3} \cdot R_{1,2}$ from (2.1) is a pseudo-random matrix from the desired uniform distribution on $SO(n)$.

Similarly, we can generate a sample from the (normalized) Haar measure μ^* on $O(n)$. Note that $O(n)$ consists of two connected components $O^+(n) = SO(n)$ and $O^-(n)$, with $O^+(n)$ being the orthonormal matrices of determinant 1 (including the identity I_n) and $O^-(n)$ being the orthonormal matrices of determinant -1. These components are isomorphic as Riemannian manifolds, and hence sampling from μ^* on $O(n)$ means obtaining independent samples from $O^+(n) = SO(n)$ and $O^-(n)$ according to the algorithm above.

4.5 Gibbs sampler on the Stiefel manifold

The Gibbs sampler iteratively draws matrices from the conditional posterior distributions of Γ and Δ. To obtain samples from both posterior distributions in one algorithm, we proceed as follows: Initialize the sampler with

1. Set $j = 0$.

2. Set a starting value for $\Delta^{(0)}$.

After initialization, each new iteration of the sampler will be:

3. Set $j = j + 1$.

4. Draw $\Gamma^{(j)}$ from $p(\Gamma|\Delta^{(j-1)}, S_d, D_\phi, A)$.

5. Draw $\Delta^{(j)}$ from $p(\Delta|\Gamma^{(j)}, S_d, D_\phi, B)$.

6. Go to step 3.

Any one of the three algorithms described in Section 4.4.2 can be used to draw values from the conditional distributions. The only approach that guarantees that the draws will be in $SO(n)$, however, is the one that relies on the Euler angle representation. This is the approach that we implement later, when analyzing the trace element concentration data introduced in Chapter 1.

4.6 D_ϕ unknown

Assume for simplicity that $m = n$. Given a sample of size d from the matrix Langevin with parameter F, let $\bar{X}$ denote the sample mean. Subject to the constraint $\bar{X}'\bar{X} = A$, the probability density function (pdf) of the eigenvalues $\lambda_1, \ldots, \lambda_n$ $(\lambda_1 > \ldots > \lambda_n > 0)$ of A given D_ϕ is ([31])

$$\frac{\pi^{n^2}}{\Gamma_n(\frac{n}{2})^2\Gamma_n(\frac{n}{2})}|D_\lambda|^{-1/2}\,{}_0F_1(\frac{n}{2}, \frac{d^2}{4}D_\phi^2, D_\lambda)\prod_{i=1}^{n-1}\prod_{j=i+1}^{n}(\lambda_i - \lambda_j)$$
$$\times \int_{U>0}|U|^{-1/2}\,{}_0F_1(\frac{n}{2}, \frac{-d^2}{4}UD_\lambda)[\,{}_0F_1(\frac{n}{2}, \frac{-1}{4}U)]^d dU,$$

where $D_\lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, the integral is taken over the set of positive-definite $n \times n$ matrices, and ${}_0F_1^{(n)}(a, A, B)$ is as in Section 4.3.2.

In principle, it might be possible to generate draws of D_ϕ by first deriving an algebraic relationship between D_λ and D_x and then using the approximations to the hypergeometric function described in Section 3.3 to generate a value for D_ϕ given a value of D_x. The approach we describe here is very difficult to implement. Therefore, in Chapter 5 we obtain a Bayesian estimate of Γ, Δ conditional on a fixed, known value of D_ϕ. To investigate whether conditioning on D_ϕ has an impact on results obtained for Γ, Δ, we carry out a sensitivity analysis by varying the value of D_ϕ over a wide range of possible values around $\hat{D}_\phi$, the MLE of D_ϕ.

5 Numerical Example: the FBI Bullet Lead Dataset

5.1 Overview

We implement the classical and Bayesian estimation methods described in Chapters 3 and 4 on a dataset obtained from the FBI. The data and some earlier analyses conducted on these data were briefly described in the Introduction chapter. Here, we discuss the data in more detail and illustrate the methods proposed earlier by modeling and drawing inferences on orthonormal matrices arising in this application. Some of the results are presented within the chapter; others, including some figures, additional tables and the code written to carry out calculations are presented in appendices later in the dissertation.

5.2 The FBI dataset on trace element concentrations in lead alloy

The data we analyze here were collected by the FBI and have been described in detail elsewhere ([35]; [7]). The dataset includes the concentration of five trace elements: antimony, copper, arsenic, bismuth, and silver in the lead alloy used in bullet tips. These trace element concentrations were measured using inductively coupled plasma

spectrometry (ICP-S) on .38-caliber cartridges loaded with 158 grain, round nose bullets. Two hundred bullets made by each of the four major U.S. manufacturers: Remington, Cascade, Federal, and Winchester were included in the study. For laboratory analyses, each bullet tip was first quartered and three of the four quarters were randomly selected. The measurements we use here are the averages (over the three quarters) of the ICP-S measurements of each of the five elements on each bullet.

Table 5.1 at the end of this chapter shows the mean and standard deviations of the five trace element concentrations by manufacturer. As is clear from the table and as shown by ([7]), manufacturers use lead alloy with varying chemical composition and it is possible to distinguish bullets made by each of the major companies by, for example, partitioning the 5-dimensional sample space using any classification method. Because here we focus on modeling orthogonal matrices and use these data simply for illustration, we ignore manufacturer information and consider the entire sample of $N = 800$ measurements together.

Table 5.1 Table of Mean Element Concentrations and Standard Deviations by Manufacturer (units are ppm v/w).

		Trace element				
		Antimony	Copper	Arsenic	Bismuth	Silver
Overall	Mean	16541.51	294.27	438.96	107.35	45.09
	Std	10678.16	129.15	557.11	65.99	17.13
Cascade	Mean	26836.06	262.16	233.42	128.44	37.60
	Std	954.82	77.14	133.71	23.59	11.41
Federal	Mean	27436.60	277.70	1381.45	16.48	65.49
	Std	367.60	22.06	114.59	0.701	8.06
Remington	Mean	7288.87	399.67	104.79	169.04	36.95
	Std	1175.05	184.84	59.01	60.35	18.36
Winchester	Mean	4604.53	237.57	36.16	115.43	40.30
	Std	569.76	103.21	34.36	25.45	9.06

We first grouped bullets into clusters using a hierarchical clustering method. In this application, we used Ward's minimum variance method ([27], [40]) to create 16 disjoint clusters. Our dataset for analysis consists of the 16 orthonormal matrices that can be applied to the observation vectors to make the clusters iso-directional.

5.3 Preliminary calculations

To build our dataset of orthonormal matrices we consider the sample of N n-dimensional observations and implement the following steps:

1. Observations are first classified into d groups using a suitable algorithm. We applied the Ward clustering algorithm ([40]) to construct the groups.

2. Within each group, observation vectors are centered around the origin by subtracting from each the group mean vector. We use X_{kj} to denote the centered kth observation vector in group j, $k = 1, \ldots, d_j$, $j = 1, \ldots, d$, and d_j is the number of observations in the jth cluster.

3. For each group j we find the $n \times n$ proper orthogonal matrix T_j such that

$$T_j = \arg\min_{T} \left\{ \sum_{i=1}^{n-1} \lambda_i ||1 - e_i(TS_j^2)||_{\text{ang}}^2 \right\}, \tag{5.1}$$

   where

   - $e_i(TS_j^2)$ is the ith eigenvector of the sample variance-covariance matrix S_j^2 of the rotated observations, i.e. the variance-covariance matrix of $X_{kj}T_j'$.

   - λ_i is the eigenvalue associated with $e_i(TS_j^2)$.

- $||1 - e_i(TS_j^2)||^2_{\text{ang}} = \arccos(\langle 1, e_i(TS_j^2)\rangle)$.

The resulting $T_1, \ldots, T_d$ are our dataset for analysis.

5.3.1 Eigenvalues and eigenvectors of $X_{kj}T_j'$

The T_j that minimizes (5.1) cannot be calculated analytically. Instead, equation (5.1) must be solved iteratively. In principle, it would appear that to calculate the T_j that minimizes (5.1) for each candidate T_j we need to rotate and center the observations, then compute the variance-covariance matrix with its eigenvalues and associated eigenvectors and finally compute the value of the objective function (5.1). These steps would need to be repeated for each candidate T_j until the minimum of (5.1) (to a desired level of accuracy) is achieved. However, we show below that finding the orthonormal matrices that minimize (5.1) does not require such a lengthy process. To derive the T_j's that minimize (5.1), it suffices to compute the eigenvalues and eigenvectors of the sample covariance matrix computed from the unrotated observations, and then apply T_j to those eigenvalues and eigenvectors, since the following holds.

Let Y denote a $n \times m$ matrix whose n rows are the m-dimensional centered observations. Further, for $X = Y/\sqrt{n-1}$ we have that $X'X$ is the variance-covariance matrix of Y. Let $R = XT'$ be the matrix of rotated observations, where T is an orthogonal matrix. The matrix T can be either proper (with determinant equal to 1) or improper (with determinant equal to -1). Then, the variance-covariance matrix of the observations after rotation is $TX'XT'$. Further, let λ be the vector of eigenvalues of $X'X$. That is, λ is the unique solution of the characteristic equation $|X'X - \lambda I| = 0$. If we denote by

λ^* the vector of eigenvalues of $TX'XT'$, we have that $\lambda = \lambda^*$ as we now show:

$$\begin{aligned} 0 = |TX'XT' - \lambda^* I| &= |TX'XT' - \lambda^* TT'| = |T(X'X - \lambda^* I)T'| \\ &= |T||X'X - \lambda^* I||T'| = |X'X - \lambda^* I| = 0. \end{aligned}$$

The equality above follows because either $|T| = |T'| = 1$ or $|T| = |T'| = -1$. Therefore, $\lambda = \lambda^*$. Further, if z_i denotes the eigenvector associated with the ith eigenvalue (λ_i) of $X'X$, $i = 1, \ldots, d$, then, Tz_i is the eigenvector associated with the ith eigenvalue of $TX'XT'$ as we show below:

$$\lambda_i = \frac{z_i' X'X z_i}{z_i' z_i} = \frac{z_i' T'TX'XT'Tz_i}{z'T'Tz_i} = \frac{(Tz_i)'TX'XT'(Tz_i)}{(Tz_i)'(Tz_i)}.$$

5.3.2 Building T

We implemented the approach described in the preceding section on the $d = 16$ clusters obtained from the $N = 800$ trace element concentration vectors. We first estimated the sample variance-covariance matrix in each cluster and using the generalized Euler angles methodology described earlier, we found the "best" rotation matrices. Here, best means that the distance measured by (5.1) between the rotated variance-covariance matrix and the identity matrix was minimized. All measurements were log transformed to reduce the skewness in the marginal distributions of the five measurements. We found, for example, that the antimony concentrations were disproportionally large relative to the concentration of the other four elements. Thus, in the original scale the first eigenvalue will always be dominated by the antimony concentrations making the contribution of all other measurements almost negligible and resulting in one very large eigenvalue and four very small ones.

The 16 rotation matrices were found and their mean was calculated. It was found to be

$$\bar{X} = \sum_{i=1}^{16} T_i = \begin{bmatrix} 0.0115967 & 0.2481433 & 0.2997253 & 0.1991757 & 0.1447478 \\ -0.0017190 & 0.3981104 & 0.0962394 & 0.0798370 & 0.3136503 \\ 0.0103284 & 0.0182594 & 0.0214524 & 0.4943292 & 0.1933737 \\ 0.0660918 & 0.2208678 & 0.1852667 & 0.0992029 & 0.2435964 \\ 0.9848410 & 0.0009513 & -0.0348590 & 0.0434607 & -0.0456850 \end{bmatrix}.$$

The 16 sample variance-covariance matrices for each group and their associated rotation matrices are given in Appendix A.

5.4 The probability model for T

We assume here that the sample of size 16 of rotation matrices $T_1, ..., T_{16}$ can be described by the matrix Langevin distribution on the Stiefel manifold that was discussed in Chapter 3. Here, $d = 16$ is the sample size and $n \times n = 5 \times 5$ is the dimensionality of the parameter space. The matrix Langevin distribution is indexed by the 5×5 parameter matrix F which in its SVD form can be written as $F = \Gamma D_\phi \Delta$ as was described in Section 3.2. In the next section, we obtain the MLE for D_ϕ, Γ and Δ following the approach described in Chapter 3. When implementing the Bayesian approach to estimation, however, we keep D_ϕ fixed at its MLE and then carry out sensitivity analyses to investigate whether the choice of a fixed value for D_ϕ critically affects inferences about Γ, Δ and by extension, for F.

5.5 Maximum likelihood estimation of F

We maximized the matrix Langevin distribution as described in Chapter 3 and obtained the maximum likelihood estimators (MLEs) of the SVD components of the parameter matrix F. To obtain MLEs of Γ, Δ, D_ϕ we first estimate the components of the SVD of $\bar{X}$ as described below. Note that because we are working on $SO(5)$, or equivalently on $V_{4,5}$, the last column of $\bar{X}$ was discarded before finding its singular value decomposition. The code used to carry out the computations is presented in Appendix B.

Using the notation introduced in Chapter 3, the singular value decomposition of $\bar{X}$ is given by $\bar{X} = H_1 D_x H_2$. MLEs of H_1, H_2 and D_x are denoted $\hat{H}_1$, $\hat{H}_2$ and $\hat{D}_x$, respectively and were calculated to be

$$\hat{H}_1 = \begin{bmatrix} 0.0375 & 0.5996 & 0.1455 & 0.6211 \\ 0.0202 & 0.5012 & 0.4384 & -0.7246 \\ 0.0553 & 0.4652 & -0.8567 & -0.2086 \\ 0.0829 & 0.4054 & 0.2291 & 0.2133 \\ 0.9941 & -0.0925 & 0.0141 & -0.0149 \end{bmatrix},$$

$$\hat{H}_2 = \begin{bmatrix} 0.9953 & -0.0768 & 0.0490 & 0.0322 \\ 0.0380 & 0.6420 & 0.5699 & -0.5114 \\ -0.0050 & 0.4550 & 0.2538 & 0.8536 \\ 0.0887 & 0.6123 & -0.7800 & -0.0939 \end{bmatrix},$$

$$\hat{D}_x = \begin{bmatrix} 0.9901 & 0 & 0 & 0 \\ 0 & 0.6951 & 0 & 0 \\ 0 & 0 & 0.4310 & 0 \\ 0 & 0 & 0 & 0.1781 \end{bmatrix}.$$

We can now compute the MLEs of the components Γ, D_ϕ, and Δ of the SVD of the parameter F of the matrix Langevin distribution following the methods in Section 3.3. Note that $\hat{\Gamma} = \hat{H}_1$, $\hat{\Delta} = \hat{H}_2$ and $\hat{D}_\phi$ is obtained using the approximation to the hypergeometric function ${}_0F_1(\frac{1}{2}m, \frac{1}{4}D_\phi)$ given in (3.3) ([31]). We used this approximation to the function because the eigenvalues of $\bar{X}$ were all relatively small. The MLEs of Γ, Δ, D_ϕ are

$$\hat{\Gamma} = \begin{bmatrix} 0.0375 & 0.5996 & 0.1455 & 0.6211 \\ 0.0202 & 0.5012 & 0.4384 & -0.7246 \\ 0.0553 & 0.4652 & -0.8567 & -0.2086 \\ 0.0829 & 0.4054 & 0.2291 & 0.2133 \\ 0.9941 & -0.0925 & 0.0141 & -0.0149 \end{bmatrix},$$

$$\hat{\Delta} = \begin{bmatrix} 0.9953 & -0.0768 & 0.0490 & 0.0322 \\ 0.0380 & 0.6420 & 0.5699 & -0.5114 \\ -0.0050 & 0.4550 & 0.2538 & 0.8536 \\ 0.0887 & 0.6123 & -0.7800 & -0.0939 \end{bmatrix},$$

$$\hat{D}_\phi = \begin{bmatrix} 4.9506 & 0 & 0 & 0 \\ 0 & 3.4754 & 0 & 0 \\ 0 & 0 & 2.1549 & 0 \\ 0 & 0 & 0 & 0.8903 \end{bmatrix}.$$

The MLE $\hat{F}$ of F is therefore

$$\hat{F} = \widehat{\Gamma D_\phi \Delta} = \hat{\Gamma}\hat{D}_\phi\hat{\Delta} = \begin{bmatrix} 0.3115 & 1.8047 & 0.8449 & -0.8441 \\ 0.1039 & 1.1456 & 1.7406 & -0.0207 \\ 0.3264 & 0.0634 & 0.6112 & -2.3764 \\ 0.4763 & 1.2139 & 0.8002 & -0.3037 \\ 4.8849 & -0.5786 & 0.0762 & 0.3499 \end{bmatrix}.$$

5.6 Bayesian estimation of F

Here, we treated the matrix of eigenvalues D_ϕ of F as known and fixed it at its MLE. To investigate whether the choice of D_ϕ critically impacts the estimated posterior distributions for Γ and Δ we conducted a sensitivity analysis. To do so, we fitted the model multiple times, each time choosing a different value for the entries in D_ϕ. Values were set by multiplying the entries in D_ϕ by 0.1, 0.5, 2, 5, and 10. We found that the estimated posterior distributions of Γ and Δ were robust to the choices of D_ϕ. We therefore fixed D_ϕ at its MLE in the analyses that follow. We note again that all inferences about Γ and Δ are conditional on $D_\phi = \hat{D}_\phi$.

Although the conditional distributions required to implement the Gibbs sampler described in Section 4.5 were expressed in terms of the matrix sum S_d we found that in

practice this is not a viable parametrization. This is due to the fact that the Gibbs sampler as described in Section 4.5 to generate random matrices from the matrix Langevin distribution requires the evaluation of the function $\text{etr}(F'S_d - D_\phi)$ at each iteration. This function is very close to 0 for almost all values of F. Hence, we re-computed the conditional distributions replacing S_d with $\bar{X} = d$.

Additionally we tested whether our algorithms to produce pseudo-random matrices from the invariant measure on $V_{m,n}$ were, in fact, producing matrices that could be regarded as pseudo-matrices from such a distribution. To do so we used a Rayleigh test as described in [30]. That is, for each algorithm, we generated a sample of size k of pseudo-random matrices and their mean was calculated. Then, the hypothesis of uniformity was tested against an alternative hypothesis of non-uniformity by calculating the Rayleigh statistic

$$S = kn\text{tr}(\bar{X}'\bar{X}) \left[1 - \frac{1}{2k}\left(1 - \frac{kn\text{tr}(\bar{X}'\bar{X})}{mn+2}\right)\right].$$

For k large enough, $S \sim \chi^2_{mn}$. Thus, we reject the null hypothesis of uniformity when $S > \chi^2_{mn;1-\alpha}$. We used $k = 1000$ and $\alpha = 0.05$. Our tests lead us to conclude that our algorithms do in fact generate pseudo-matrices from the Haar invariant measure on $V_{m,n}$.

5.6.1 Gibbs sampler

We simulated four parallel chains of length 20,000 from the marginal posterior distributions of Γ and Δ, with D_ϕ fixed at its MLE. We discarded the first 10,000 draws from each chain as burn-ins. Convergence to the target distributions was assessed using

the convergence diagnostics proposed by Gelman and Rubin ([18]), and by Raftery and Lewis ([36]). We also monitored the behavior of autocorrelations and cross-correlations as recommended by Cowles and Carlin ([10]). Based on those diagnostics, we are comfortable that after 20,000 iterations our draws appear to have converged to the desired stationary distribution. Trace plots for all parameters (entries in the Γ and Δ matrices) and of diagnostic statistics are shown in Appendix C.

Although we programmed the three algorithms that were described in Section 4.4.2, simulations were actually conducted using the generalized Euler angles algorithm. We chose this approach because the draws of Δ are restricted to $SO(n)$. The generalized Euler angles parameterization of orthogonal matrices automatically produces a matrix with the desired properties whereas the algorithms proposed by Chikuse ([9]) and Hoff ([23]) do not. These algorithms require a "correction" of the matrices that are generated and the additional steps decrease computational efficiency. For example, we implemented the algorithm proposed by Chikuse ([9]) 10,000 times and found that on the average 18 matrices have to be generated before returning a proper 4×4 orthogonal matrix. The average number of matrices needed before a member of $SO(5)$ was found was 29. As the dimension of $SO(n)$ increases, so does the number of matrices that need to be generated before the desired matrix is returned by the algorithm. The algorithm proposed by Hoff ([23]) is also not guaranteed to result in proper $SO(n)$ matrices. Using the same procedure described above we found that the average number of matrices needed before a member of $SO(n)$ is generated is 5 for $n = 4$, and 6 for $n = 5$. Hence, this algorithm is more efficient than that of Chikuse's in terms of the number of matrices that it needs

to generate to obtain a matrix from the appropriate space. However, the approach proposed by Hoff is more computer intensive than the one proposed by Chikuse due to the need to solve systems of linear equations in each step for the computation of the null spaces. In the end, it requires more CPU time to produce the same rotation matrix.

Table 5.2 shows the mean and selected percentiles of the estimated posterior distribution of Γ after combining the four chains. Table 5.3 shows the same summary posterior statistics for Δ.

Table 5.4 and Figure 5.1 show the matrix Δ in terms of its generalized Euler angles. Table 5.5 and Figure 5.2 show the characterization of Γ via its angles. Angles are measured in degrees.

Finally, Table 5.6 shows a summary of the posterior distribution of the matrix of parameters F. Those values were calculated by rebuilding F at each iteration of the Gibbs sampler, with D_ϕ fixed at its MLE.

Table 5.2 Summary of the posterior distribution of Γ.

		Posterior quantiles				
Estimand	Mean	2.5%	25%	Median	75%	97.5%
$[1,1]$	-0.03215	-0.70039	-0.22508	-0.01502	0.15529	0.62715
$[1,2]$	0.04529	-0.28461	-0.06149	-0.00037	0.11214	0.59724
$[1,3]$	-0.47477	-0.95680	-0.75345	-0.53188	-0.22367	0.20670
$[1,4]$	-0.19557	-0.98839	-0.75312	-0.25400	0.29662	0.90729
$[2,1]$	0.13799	-0.64183	-0.08583	0.14402	0.39588	0.79308
$[2,2]$	0.03645	-0.63336	-0.14479	0.02122	0.21692	0.72263
$[2,3]$	0.17084	-0.61235	-0.02824	0.19682	0.41663	0.76326
$[2,4]$	-0.01565	-0.66741	-0.24406	0.02593	0.22107	0.51832
$[3,1]$	-0.31751	-0.91747	-0.65480	-0.39581	-0.04994	0.66450
$[3,2]$	0.02585	-0.84634	-0.36917	0.04305	0.41790	0.87937
$[3,3]$	-0.03503	-0.72938	-0.31078	-0.03030	0.23949	0.66842
$[3,4]$	0.06543	-0.59273	-0.15429	0.04969	0.27930	0.74551
$[4,1]$	0.03350	-0.84451	-0.39073	0.03210	0.46093	0.90176
$[4,2]$	-0.36977	-0.96364	-0.75487	-0.48001	-0.07267	0.72343
$[4,3]$	0.20368	-0.60337	-0.02251	0.20983	0.47252	0.83940
$[4,4]$	0.04416	-0.66807	-0.16928	0.04257	0.25219	0.76373
$[5,1]$	-0.03640	-0.85063	-0.38321	-0.03493	0.31317	0.78684
$[5,2]$	0.04519	-0.84606	-0.34682	0.02841	0.45000	0.92010
$[5,3]$	0.13706	-0.70229	-0.17282	0.14262	0.46932	0.88100
$[5,4]$	-0.40972	-0.93396	-0.67725	-0.41497	-0.13445	0.12146

$[i,j]$ represents the element of the ith row and jth column

Table 5.3 Summary of the posterior distribution of Δ.

		Posterior quantiles				
Estimand	Mean	2.5%	25%	Median	75%	97.5%
$[1,1]$	-0.00469	-0.85605	-0.32502	-0.00369	0.31068	0.85706
$[1,2]$	-0.25781	-0.90513	-0.49744	-0.18621	-0.00806	0.24728
$[1,3]$	0.17340	-0.96678	-0.28284	0.29108	0.68235	0.97435
$[1,4]$	-0.40115	-0.98407	-0.61989	-0.34517	-0.15189	-0.01360
$[2,1]$	0.00342	-0.86147	-0.35154	0.00069	0.35932	0.86876
$[2,2]$	0.08276	-0.79882	-0.17946	0.10261	0.38298	0.81924
$[2,3]$	-0.01318	-0.70357	-0.31189	-0.01501	0.25759	0.76742
$[2,4]$	-0.54895	-0.98377	-0.83975	-0.67031	-0.33483	0.41888
$[3,1]$	0.01742	-0.91600	-0.45512	0.02495	0.49087	0.92152
$[3,2]$	0.23023	-0.81479	-0.19114	0.31986	0.69406	0.96418
$[3,3]$	-0.03516	-0.86898	-0.37167	-0.02953	0.27947	0.84251
$[3,4]$	-0.19623	-0.86247	-0.46488	-0.14522	0.04962	0.37835
$[4,1]$	-0.01658	-0.90364	-0.44410	-0.02023	0.40615	0.89714
$[4,2]$	-0.08270	-0.94840	-0.56208	-0.11852	0.37054	0.92975
$[4,3]$	0.28938	-0.64461	0.00484	0.32210	0.62555	0.93398
$[4,4]$	0.14386	-0.66277	-0.05074	0.11970	0.38623	0.85434

$[i,j]$ represents the element of the ith row and jth column

Table 5.4 Summary of the posterior distributions of the angles that determine Δ.

		Posterior quantiles				
Estimand	Mean	2.5%	25%	Median	75%	97.5%
$\alpha^{\Delta}_{1,2}$	309.534	197.068	287.270	321.876	343.760	358.538
$\alpha^{\Delta}_{1,3}$	258.155	129.568	220.976	266.151	303.510	345.029
$\alpha^{\Delta}_{2,3}$	207.638	80.211	162.288	210.459	255.845	318.593
$\alpha^{\Delta}_{1,4}$	156.334	42.560	108.544	154.123	202.074	281.658
$\alpha^{\Delta}_{2,4}$	104.168	15.808	58.509	96.530	142.212	231.849
$\alpha^{\Delta}_{3,4}$	26.367	0.779	8.736	20.192	38.322	83.787

Table 5.5 Summary of the posterior distributions of the angles that determine Γ.

		Posterior quantiles				
Estimand	Mean	2.5%	25%	Median	75%	97.5%
$\alpha^{\Gamma}_{1,2}$	326.783	248.362	313.129	335.067	349.259	359.054
$\alpha^{\Gamma}_{1,3}$	293.832	198.196	270.767	300.538	324.094	350.415
$\alpha^{\Gamma}_{2,3}$	261.154	157.959	232.632	266.493	295.563	334.882
$\alpha^{\Gamma}_{1,4}$	229.050	124.570	196.116	232.814	265.405	314.710
$\alpha^{\Gamma}_{2,4}$	196.763	95.060	160.868	199.130	233.753	289.691
$\alpha^{\Gamma}_{3,4}$	164.118	68.307	127.183	163.906	200.703	262.242
$\alpha^{\Gamma}_{1,5}$	131.757	44.729	94.698	129.046	165.875	233.366
$\alpha^{\Gamma}_{2,5}$	99.514	24.712	64.597	94.764	129.862	198.759
$\alpha^{\Gamma}_{3,5}$	66.975	9.594	36.086	60.283	91.250	159.823
$\alpha^{\Gamma}_{4,5}$	16.752	0.493	5.348	12.559	24.039	55.496

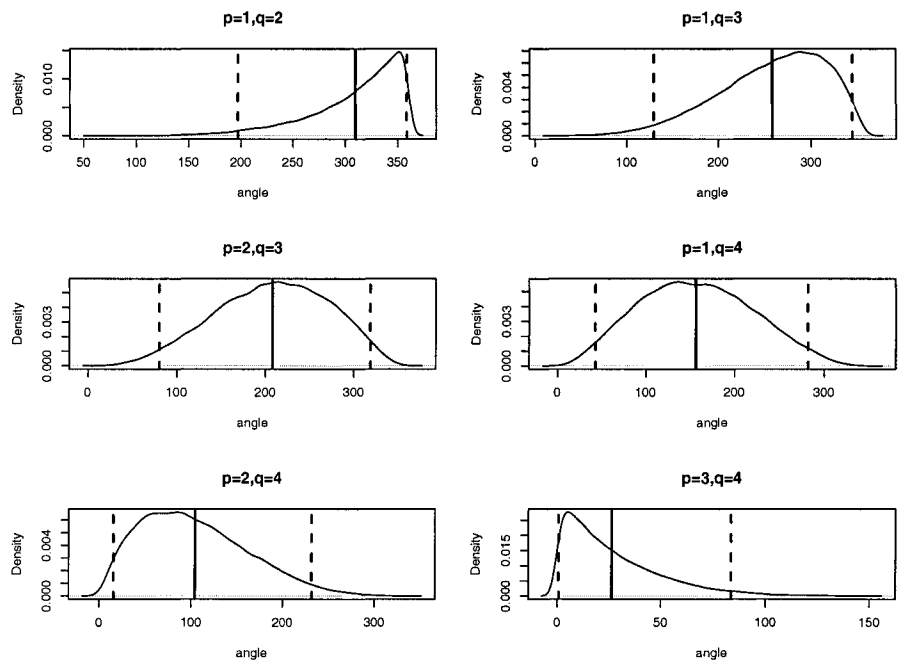

Figure 5.1 Posterior distribution of the angles $\alpha^{\Delta}_{p,q}$ that determine Δ.

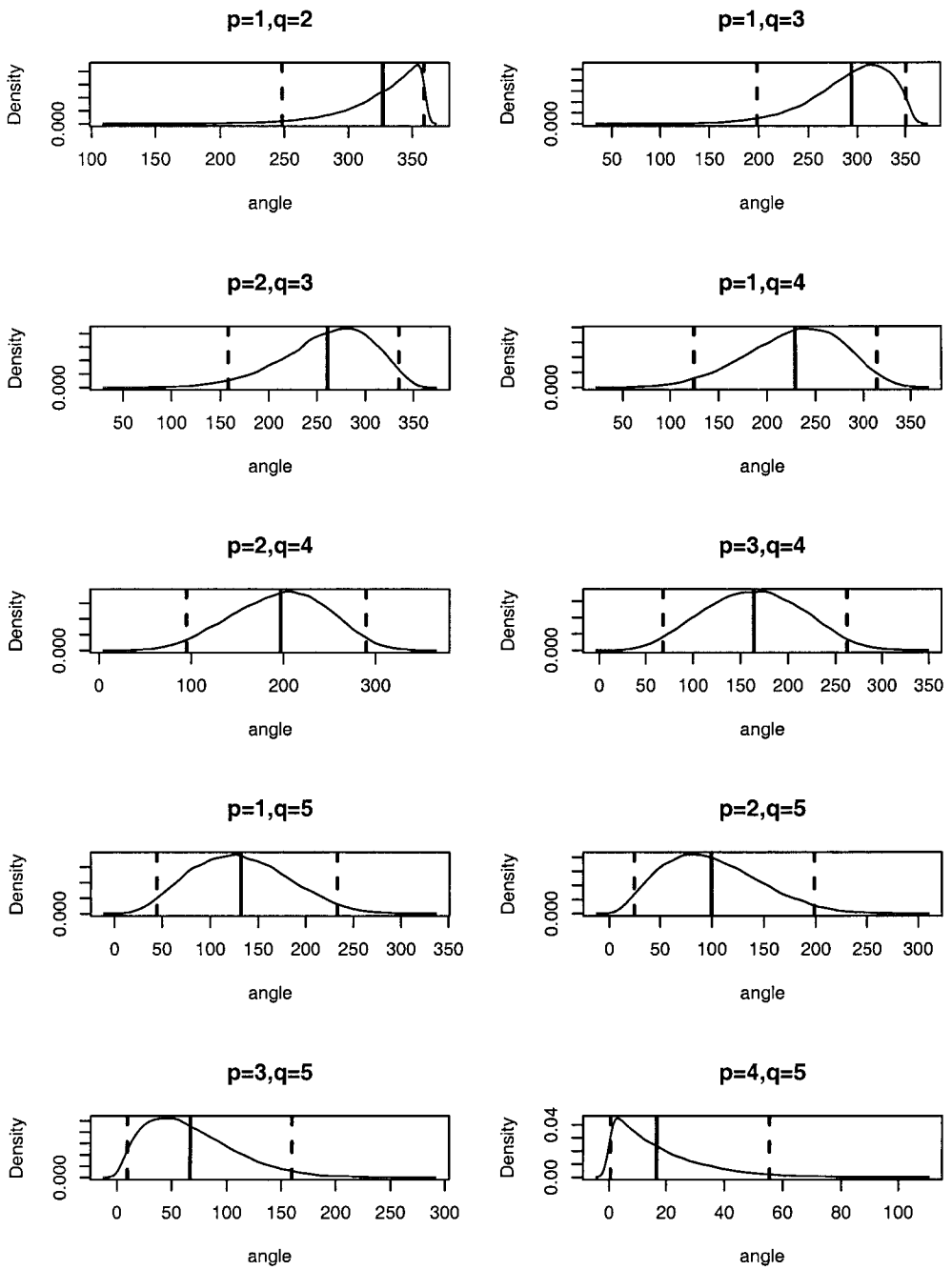

Figure 5.2 Posterior distribution of the angles $\alpha_{p,q}^{\Gamma}$ that determine Γ.

Table 5.6 Summary of the posterior distribution of F.

		Posterior quantiles				
Estimand	Mean	2.5%	25%	Median	75%	97.5%
[1, 1]	-0.00422	0.13973	-0.00357	-0.14727	-0.41911	0.40568
[1, 2]	-0.02871	0.11042	-0.03673	-0.17227	-0.39366	0.38669
[1, 3]	0.00777	0.15414	-0.00055	-0.14052	-0.45177	0.48509
[1, 4]	0.03202	0.18773	0.03779	-0.10996	-0.46162	0.47061
[2, 1]	-0.00060	0.13108	-0.00162	-0.13358	-0.46083	0.46817
[2, 2]	-0.00498	0.11815	0.00358	-0.11918	-0.45939	0.40352
[2, 3]	0.02280	0.17485	0.01400	-0.13074	-0.52463	0.58226
[2, 4]	-0.08532	0.05587	-0.07664	-0.23279	-0.57500	0.40728
[3, 1]	0.00555	0.21343	0.00489	-0.20046	-0.58477	0.59186
[3, 2]	0.07396	0.24544	0.05917	-0.10558	-0.43908	0.63249
[3, 3]	-0.05102	0.19056	-0.05185	-0.30445	-0.71725	0.66815
[3, 4]	0.13391	0.38091	0.14435	-0.09962	-0.51126	0.71638
[4, 1]	0.01044	0.23971	0.01506	-0.21787	-0.59695	0.60652
[4, 2]	-0.00071	0.20880	0.00868	-0.20203	-0.59942	0.55104
[4, 3]	0.03459	0.30416	0.02901	-0.23328	-0.65940	0.73463
[4, 4]	0.11530	0.39594	0.14877	-0.13605	-0.60998	0.68212
[5, 1]	0.07936	0.27583	0.08568	-0.11578	-0.47431	0.61623
[5, 2]	0.03296	0.21285	0.03680	-0.14334	-0.50132	0.54757
[5, 3]	-0.03122	0.19082	-0.04242	-0.25225	-0.65286	0.61678
[5, 4]	-0.02134	0.20902	-0.02513	-0.25558	-0.62400	0.60297

$[i, j]$ represents the element of the ith row and jth column

6 Discussion

We have discussed classical and Bayesian estimation of the parameters of the matrix Langevin distribution on the Stiefel manifold. This probability model is an appropriate model for orthogonal rotational matrices arising in many applications. The matrix Langevin distribution has been characterized elsewhere (e.g., [26], [31]), as has maximum likelihood estimation in the lower-dimensional von Mises and von Mises-Fisher distributions ([28], [30]) and in the $n-$dimensional Langevin distribution ([9]). Further, ([33]) describes a Bayesian approach for parameter estimation in the von Mises distribution. The literature on Bayesian estimation of the Langevin parameters is scarce, however, and focused on special cases and specific applications (e.g., [1], [14]).

In this dissertation we have extended some of the results presented by Chikuse ([9]) on the properties of maximum likelihood estimators of the parameters of the matrix Langevin distribution and have proposed a Bayesian approach to estimation. We have used the Haar measure as our reference measure, justification for this choice can be found elsewhere (e.g., [20], [25]). Because the Bayesian methods must be implemented numerically, we have proposed an algorithm for drawing orthogonal matrices from the appropriate conditional distributions that greatly improve the efficiency of algorithms proposed earlier by Chikuse ([9]) and Hoff ([23]).

One unresolved problem in this dissertation has to do with the choice of prior distributions for the components in the singular value decomposition (SVD) of the matrix parameter F in the Langevin distribution. To construct the joint prior distribution of the matrices Γ and Δ described earlier, we assume that these matrices are independent a priori. This is not a justifiable assumption since for a given F, Γ and Δ are algebraically dependent. A more plausible joint prior distribution would be constructed taking into account this dependence.

Bayesian estimates of Γ and Δ in Chapter 4 were obtained assuming that the diagonal matrix D_ϕ in the SVD of F is known. Relaxing this condition appears to be quite challenging, as we discuss in Section 4.7 as the required algebraic and numerical procedures may be very difficult to implement. To investigate the impact of conditioning on D_ϕ we carried out a sensitivity analysis that suggested that inferences about Γ and Δ are robust to specifications of D_ϕ. Yet finding the means to include D_ϕ among the parameters to be estimated jointly, at least in the case of rotation matrices in the special $SO(n)$ group would be an important contribution to this area.

We applied the methodology described in Chapters 3 and 4 to a dataset obtained from the FBI in the context of an earlier collaboration. Given a sample of orthogonal rotation matrices we fitted a matrix Langevin model from a classical and a Bayesian perspective. We did not, however, carry out any diagnostic tests to investigate whether the model is plausible for the sample. To determine whether the model fits the data well, one might consider carrying out posterior predictive checks as described, for example, in Chapter 6 of Gelman et al. ([19]). Typically, posterior predictive tests are carried out

as follows. A number M of replicate samples of size d of orthogonal matrices are first generated from the model by randomly drawing M values of the parameters from the joint posterior distribution. A discrepancy measure $A(X, F)$ that may depend on both the sample and the parameters is then calculated from the original sample and from the replicated samples. The distribution (over the possible values of F) of $A(X, F)$ in the original sample is then compared to the distribution (over the replicate samples and the possible values of F) of $A(X, F)$ in the replications. A posterior predictive p-value can be computed by comparing the tail area probabilities of the distribution of the discrepancy statistic in the original sample and in the replicated samples. Very small or very large values of p indicate that the model does not fit the data well.

Some of the results we present in this work suggest potentially useful extensions of pursuit algorithms in visualization packages such as Gobi ([39]). Standard pursuit algorithms rely on the Euclidean geometry. Here, we present results that permit calculating distances in Riemannian spaces of non-zero curvature. If it is known that observations do not reside in spaces of zero curvature, then it might be possible to greatly increase the efficiency of pursuit algorithms by using some of our results to carry out distance calculations. These models can be used as one more possible model in grand tours when we do not know the geometry of the space of the observations.

APPENDIX A Data

For cluster j, $j = 1, \ldots, 16$: S_j denotes the variance-covariance matrix and T_j denotes the matrix of rotation that minimizes (5.1).

Cluster 1

$$S_1 = \begin{bmatrix} 0.00075916 & -.00064933 & -.00893661 & -.00283945 & 0.00283120 \\ -.00064933 & 0.03239143 & 0.02857120 & 0.00697672 & -.01008371 \\ -.00893661 & 0.02857120 & 0.49429016 & 0.08655390 & -.09804371 \\ -.00283945 & 0.00697672 & 0.08655390 & 0.02750590 & -.01672780 \\ 0.00283120 & -.01008371 & -.09804371 & -.01672780 & 0.03232953 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} -.01825289 & 0.06149926 & 0.96309219 & 0.17276635 & -.19618843 \\ -.01129125 & 0.97278234 & -.11285235 & 0.06396527 & -.19167633 \\ -.02498616 & 0.07360631 & -.01259816 & 0.76594965 & 0.63805973 \\ 0.14381587 & 0.21029251 & 0.24329012 & -.60578997 & 0.71338803 \\ 0.98905613 & -.01647812 & -.01920901 & 0.11135474 & -.09342152 \end{bmatrix}$$

Cluster 2

$$S_2 = \begin{bmatrix} 0.00006669 & -.00011416 & 0.00008029 & -.00000456 & 0.00010014 \\ -.00011416 & 0.00543921 & 0.00059729 & 0.00091218 & -.00038331 \\ 0.00008029 & 0.00059729 & 0.00041748 & 0.00028055 & 0.00013189 \\ -.00000456 & 0.00091218 & 0.00028055 & 0.00059501 & 0.00001469 \\ 0.00010014 & -.00038331 & 0.00013189 & 0.00001469 & 0.00064646 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} -.01942862 & 0.97390248 & 0.11740018 & 0.17994178 & -.07053119 \\ 0.16646425 & -.07233067 & 0.48009049 & 0.39489703 & 0.76198910 \\ -.03887907 & -.21262726 & 0.27722905 & 0.74196940 & -.57087961 \\ 0.27343425 & -.01595252 & 0.77003214 & -.49928348 & -.28765559 \\ 0.94637958 & 0.02859029 & -.29312946 & 0.10897114 & -.07582015 \end{bmatrix}$$

Cluster 3

$$S_3 = \begin{bmatrix} 0.00267596 & 0.00978171 & -.00365174 & -.00229733 & -.00136640 \\ 0.00978171 & 0.25084716 & 0.02308903 & -.00472973 & 0.02408132 \\ -.00365174 & 0.02308903 & 0.66420611 & 0.02640949 & -.03976648 \\ -.00229733 & -.00472973 & 0.02640949 & 0.03170424 & -.01067576 \\ -.00136640 & 0.02408132 & -.03976648 & -.01067576 & 0.03871695 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} -.00472984 & 0.05086175 & 0.99591043 & 0.04194353 & -.06159444 \\ 0.03890296 & 0.99058596 & -.04161005 & -.03229145 & 0.12021594 \\ -.02436437 & -.10838085 & 0.07914528 & -.63955065 & 0.75655202 \\ -.14567329 & -.04568371 & 0.00862407 & 0.76065922 & 0.63088467 \\ 0.98825597 & -.04815724 & 0.00962693 & 0.09782901 & 0.10662002 \end{bmatrix}$$

Cluster 4

$$S_4 = \begin{bmatrix} 0.00111547 & -.00534186 & 0.00561374 & -.00140197 & 0.00069554 \\ -.00534186 & 0.04541709 & -.00316498 & 0.01720066 & 0.00349709 \\ 0.00561374 & -.00316498 & 0.06535076 & 0.00669972 & 0.01283593 \\ -.00140197 & 0.01720066 & 0.00669972 & 0.00851057 & 0.00334901 \\ 0.00069554 & 0.00349709 & 0.01283593 & 0.00334901 & 0.00361195 \end{bmatrix}$$

$$T_4 = \begin{bmatrix} 0.08336649 & -.04474865 & 0.97102943 & 0.10318737 & 0.19365383 \\ -.10372709 & 0.92147190 & -.00466397 & 0.36406453 & 0.08697987 \\ 0.04089123 & -.33559321 & -.20846381 & 0.72349093 & 0.56463157 \\ 0.11972932 & 0.16511272 & -.09880200 & -.57319803 & 0.78745467 \\ 0.98299721 & 0.09487934 & -.06213778 & 0.06938490 & -.12664528 \end{bmatrix}$$

Cluster 5

$$S_5 = \begin{bmatrix} 0.00038142 & -.00072102 & 0.00928136 & 0.00569698 & 0.00454360 \\ -.00072102 & 0.02751849 & 0.00993933 & 0.01127771 & 0.00765939 \\ 0.00928136 & 0.00993933 & 0.36801435 & 0.22987793 & 0.17704030 \\ 0.00569698 & 0.01127771 & 0.22987793 & 0.14736587 & 0.11333152 \\ 0.00454360 & 0.00765939 & 0.17704030 & 0.11333152 & 0.08759493 \end{bmatrix}$$

$$T_5 = \begin{bmatrix} 0.01964010 & 0.02836169 & 0.78168371 & 0.49335361 & 0.37997720 \\ -.03799292 & 0.98211230 & -.13345120 & 0.11581077 & 0.05282631 \\ 0.04162231 & -.17603400 & -.60786795 & 0.56554172 & 0.52719895 \\ 0.43680955 & 0.05653813 & 0.03848591 & -.59711416 & 0.66930888 \\ 0.89757217 & 0.02159919 & -.01329442 & 0.25847112 & -.35624936 \end{bmatrix}$$

Cluster 6

$$S_6 = \begin{bmatrix} 0.00038490 & 0.00263621 & -.01717160 & 0.00009288 & -.00490127 \\ 0.00263621 & 0.16087651 & -.48252346 & 0.01395393 & -.10161807 \\ -.01717160 & -.48252346 & 1.98004755 & 0.01938515 & 0.52561991 \\ 0.00009288 & 0.01395393 & 0.01938515 & 0.13883129 & -.08288247 \\ -.00490127 & -.10161807 & 0.52561991 & -.08288247 & 0.36423692 \end{bmatrix}$$

$$T_6 = \begin{bmatrix} -.00798895 & -.22893101 & 0.93441924 & -.00363706 & 0.27271636 \\ -.00037436 & 0.11180793 & -.21078005 & -.53738081 & 0.80888348 \\ -.00673014 & 0.46829768 & -.01104850 & 0.76555002 & 0.44097968 \\ -.05554338 & 0.84465877 & 0.28595569 & -.35344693 & -.27707598 \\ 0.99840156 & 0.04835715 & 0.02323184 & -.01473315 & -.00995626 \end{bmatrix}$$

Cluster 7

$$S_7 = \begin{bmatrix} 0.00021926 & 0.00078926 & 0.00034733 & -.00040895 & -.00018265 \\ 0.00078926 & 0.03733220 & 0.00980251 & 0.00053982 & -.01475904 \\ 0.00034733 & 0.00980251 & 0.00654766 & 0.00059542 & -.00780036 \\ -.00040895 & 0.00053982 & 0.00059542 & 0.01501741 & -.00651199 \\ -.00018265 & -.01475904 & -.00780036 & -.00651199 & 0.01486042 \end{bmatrix}$$

$$T_7 = \begin{bmatrix} 0.01651330 & 0.83970325 & 0.28095950 & 0.10582667 & -.45220376 \\ -.03027400 & -.32271267 & 0.02774790 & 0.86566012 & -.38052942 \\ -.02112014 & 0.43587370 & -.56340628 & 0.42511921 & 0.55804577 \\ 0.08353301 & 0.02238540 & 0.77225388 & 0.24193059 & 0.58104618 \\ 0.99568417 & -.01637094 & -.08055510 & 0.01328617 & -.04098016 \end{bmatrix}$$

Cluster 8

$$S_8 = \begin{bmatrix} 0.00005269 & 0.00033511 & 0.00017844 & 0.00001136 & -.00058457 \\ 0.00033511 & 0.07487257 & -.05032063 & 0.03367530 & -.00645207 \\ 0.00017844 & -.05032063 & 0.07366293 & -.04153720 & -.02849458 \\ 0.00001136 & 0.03367530 & -.04153720 & 0.03035297 & 0.01327300 \\ -.00058457 & -.00645207 & -.02849458 & 0.01327300 & 0.05259587 \end{bmatrix}$$

$$T_8 = \begin{bmatrix} -.00023843 & 0.58977334 & -.66465880 & 0.41010301 & 0.20545451 \\ -.01083167 & -.53177611 & -.19615975 & 0.03448883 & 0.82306058 \\ -.00192173 & 0.60735887 & 0.49258445 & -.33759280 & 0.52393051 \\ 0.01080571 & -.02188626 & 0.52636607 & 0.84650468 & 0.07597901 \\ 0.99988107 & -.00421623 & -.00702519 & -.00932560 & 0.00915104 \end{bmatrix}$$

Cluster 9

$$S_9 = \begin{bmatrix} 0.00003549 & 0.00002438 & 0.00005297 & 0.00008288 & 0.00007148 \\ 0.00002438 & 0.00015620 & 0.00004078 & 0.00012559 & 0.00000578 \\ 0.00005297 & 0.00004078 & 0.00044474 & 0.00013044 & -.00002747 \\ 0.00008288 & 0.00012559 & 0.00013044 & 0.00134251 & 0.00032049 \\ 0.00007148 & 0.00000578 & -.00002747 & 0.00032049 & 0.00087002 \end{bmatrix}$$

$$T_9 = \begin{bmatrix} 0.07485896 & 0.08714822 & 0.10235750 & 0.88788491 & 0.43357196 \\ 0.01991994 & -.09344096 & -.26596582 & -.38983451 & 0.87644921 \\ 0.12319816 & 0.05786607 & 0.94660502 & -.21944754 & 0.19301668 \\ 0.10836399 & 0.98278012 & -.10351286 & -.10539662 & 0.02402329 \\ 0.98340164 & -.12028600 & -.10958624 & -.02058562 & -.07758592 \end{bmatrix}$$

Cluster 10

$$S_{10} = \begin{bmatrix} 0.00005537 & 0.00099510 & 0.00027801 & -.00013851 & -.00047774 \\ 0.00099510 & 0.07586960 & 0.01072874 & -.00494108 & -.03081831 \\ 0.00027801 & 0.01072874 & 0.04762206 & -.01256131 & -.03018713 \\ -.00013851 & -.00494108 & -.01256131 & 0.01496289 & 0.00199267 \\ -.00047774 & -.03081831 & -.03018713 & 0.00199267 & 0.03676015 \end{bmatrix}$$

$$T_{10} = \begin{bmatrix} 0.01053860 & 0.72627658 & 0.43337734 & -.11165485 & -.52165940 \\ 0.00561783 & 0.62249766 & -.72777716 & 0.18146733 & 0.22332740 \\ -.00468570 & -.18228192 & -.11032258 & 0.82572431 & -.52226396 \\ -.00213764 & 0.22728743 & 0.51995520 & 0.52226020 & 0.63657421 \\ 0.99991542 & -.01152025 & 0.00011588 & 0.00514317 & 0.00315680 \end{bmatrix}$$

Cluster 11

$$S_{11} = \begin{bmatrix} 0.00003924 & -.00016983 & -.00073507 & 0.00027763 & 0.00049957 \\ -.00016983 & 0.04235298 & -.06228440 & 0.01889681 & 0.01635618 \\ -.00073507 & -.06228440 & 0.20943199 & -.05797563 & -.09031599 \\ 0.00027763 & 0.01889681 & -.05797563 & 0.02207471 & 0.02262107 \\ 0.00049957 & 0.01635618 & -.09031599 & 0.02262107 & 0.04664003 \end{bmatrix}$$

$$T_{11} = \begin{bmatrix} 0.00294838 & 0.26519229 & -.85792688 & 0.24107303 & 0.36811631 \\ -.01473706 & 0.86660937 & 0.08328106 & 0.08331534 & -.48465847 \\ 0.01821283 & -.26079694 & 0.04749370 & 0.91576138 & -.30129501 \\ 0.01729607 & 0.33245351 & 0.50471375 & 0.30961278 & 0.73387973 \\ 0.99957154 & 0.01099380 & -.00584025 & -.02152586 & -.01544020 \end{bmatrix}$$

Cluster 12

$$S_{12} = \begin{bmatrix} 0.00040619 & 0.00184330 & 0.00614253 & 0.00245793 & -.00619851 \\ 0.00184330 & 0.02929691 & -.02025347 & 0.02750348 & -.07790485 \\ 0.00614253 & -.02025347 & 0.40560668 & 0.01553888 & 0.00988788 \\ 0.00245793 & 0.02750348 & 0.01553888 & 0.03638574 & -.09375762 \\ -.00619851 & -.07790485 & 0.00988788 & -.09375762 & 0.26127362 \end{bmatrix}$$

$$T_{12} = \begin{bmatrix} 0.01322126 & -.07201436 & 0.99253621 & 0.01182663 & 0.09680460 \\ -.02364330 & -.26703458 & -.10277770 & -.33120097 & 0.89881933 \\ 0.04843964 & 0.95831835 & 0.04258173 & -.04311247 & 0.27496846 \\ -.06097006 & -.04616497 & -.04557042 & 0.94108401 & 0.32624480 \\ 0.99659543 & -.05478325 & -.02046332 & 0.05165512 & 0.02663364 \end{bmatrix}$$

Cluster 13

$$S_{13} = \begin{bmatrix} 0.00004177 & -.00005425 & -.00004158 & -.00005999 & 0.00009007 \\ -.00005425 & 0.00457997 & 0.00471741 & 0.00044997 & -.00013875 \\ -.00004158 & 0.00471741 & 0.00514480 & 0.00045873 & -.00033538 \\ -.00005999 & 0.00044997 & 0.00045873 & 0.00177649 & 0.00076440 \\ 0.00009007 & -.00013875 & -.00033538 & 0.00076440 & 0.01545883 \end{bmatrix}$$

$$T_{13} = \begin{bmatrix} 0.00581583 & -.02910629 & -.04315297 & 0.05305073 & 0.99721735 \\ -.00710364 & 0.68289615 & 0.72389108 & 0.08602990 & 0.04672200 \\ -.03610865 & -.05445108 & -.06346251 & 0.99420972 & -.05701567 \\ 0.11597933 & -.72262152 & 0.68135248 & 0.00854420 & 0.00726197 \\ 0.99255261 & 0.08751509 & -.07649077 & 0.03547541 & -.00843154 \end{bmatrix}$$

Cluster 14

$$S_{14} = \begin{bmatrix} 0.00002823 & -.00014376 & -.00014137 & 0.00001340 & 0.00010813 \\ -.00014376 & 0.00688426 & 0.00713430 & 0.00058385 & -.00025569 \\ -.00014137 & 0.00713430 & 0.00768474 & 0.00083799 & -.00020638 \\ 0.00001340 & 0.00058385 & 0.00083799 & 0.00168105 & 0.00100423 \\ 0.00010813 & -.00025569 & -.00020638 & 0.00100423 & 0.01539839 \end{bmatrix}$$

$$T_{14} = \begin{bmatrix} 0.00991573 & -.16674175 & -.17156030 & 0.05288021 & 0.96946873 \\ -.01167541 & 0.66350841 & 0.70427378 & 0.09490365 & 0.23369214 \\ 0.01451042 & -.13029733 & 0.01432656 & 0.98850362 & -.07394184 \\ -.04882300 & 0.71612211 & -.68832253 & 0.10479816 & -.00385676 \\ 0.99858456 & 0.04631956 & -.02392389 & -.00865560 & -.00600841 \end{bmatrix}$$

Cluster 15

$$S_{15} = \begin{bmatrix} 0.00048173 & 0.00182747 & -.00246975 & 0.00440175 & 0.00469904 \\ 0.00182747 & 0.13156123 & -.06280773 & 0.10403667 & 0.09059462 \\ -.00246975 & -.06280773 & 0.23079481 & -.12523113 & -.04687788 \\ 0.00440175 & 0.10403667 & -.12523113 & 0.19261327 & 0.13178713 \\ 0.00469904 & 0.09059462 & -.04687788 & 0.13178713 & 0.13049847 \end{bmatrix}$$

$$T_{15} = \begin{bmatrix} 0.01458698 & 0.40860397 & -.52658475 & 0.61028826 & 0.42788639 \\ 0.00949653 & 0.32310077 & 0.80786158 & 0.15177272 & 0.46887143 \\ 0.02867126 & -.84390286 & 0.09178849 & 0.44899081 & 0.27746718 \\ -.06745289 & 0.12614631 & 0.24787730 & 0.63445647 & -.71802441 \\ 0.99715851 & 0.02374353 & 0.01413792 & 0.01963500 & -.06727354 \end{bmatrix}$$

Cluster 16

$$S_{16} = \begin{bmatrix} 0.00003555 & -.00026615 & -.00030093 & 0.00001647 & 0.00016265 \\ -.00026615 & 0.00907851 & 0.00888819 & -.00002898 & -.00515889 \\ -.00030093 & 0.00888819 & 0.00911819 & -.00017456 & -.00533879 \\ 0.00001647 & -.00002898 & -.00017456 & 0.00227769 & 0.00187932 \\ 0.00016265 & -.00515889 & -.00533879 & 0.00187932 & 0.01874551 \end{bmatrix}$$

$$T_{16} = \begin{bmatrix} -.01521954 & 0.48051275 & 0.48672200 & -.06202334 & -.72672599 \\ -.01624839 & 0.51968799 & 0.50872250 & 0.13172439 & 0.67343187 \\ 0.00850390 & -.00480571 & -.07134659 & 0.98816023 & -.13547553 \\ 0.12830239 & 0.70241662 & -.69843087 & -.04837498 & -.00189048 \\ 0.99144871 & -.07496463 & 0.10680399 & -.00100888 & 0.00128738 \end{bmatrix}$$

APPENDIX B Computer code

All computations were carried out using the SAS System version 9.13.

```
%let whereis = C:\Thesis\Programs\MySASfiles;
libname thesis "&whereis";
options nodate linesize = 72;
proc iml;
* Settings: I(d) = reference matrix;
* Read data;
* The following modules define rotations about one angle;
start R45(t,a);
      t = I(5);
      t[4,4] = cos(a);  t[4,5] = sin(a);
      t[5,4] = -sin(a);  t[5,5] = cos(a);
finish;
start R35(t,a);
      t = I(5);
      t[3,3] = cos(a);  t[3,5] = sin(a);
      t[5,3] = -sin(a);  t[5,5] = cos(a);
finish;
start R25(t,a);
      t = I(5);
      t[2,2] = cos(a);  t[2,5] = sin(a);
      t[5,2] = -sin(a);  t[5,5] = cos(a);
finish;
start R15(t,a);
      t = I(5);
      t[1,1] = cos(a);  t[1,5] = sin(a);
      t[5,1] = -sin(a);  t[5,5] = cos(a);
finish;
start R34(t,a);
      t = I(5);
      t[3,3] = cos(a);  t[3,4] = sin(a);
      t[4,3] = -sin(a);  t[4,4] = cos(a);
```

```
finish;
start R24(t,a);
      t = I(5);
      t[2,2] = cos(a);  t[2,4] = sin(a);
      t[4,2] = -sin(a);  t[4,4] = cos(a);
finish;
start R14(t,a);
      t = I(5);
      t[1,1] = cos(a);  t[1,4] = sin(a);
      t[4,1] = -sin(a);  t[4,4] = cos(a);
finish;
start R23(t,a);
      t = I(5);
      t[2,2] = cos(a);  t[2,3] = sin(a);
      t[3,2] = -sin(a);  t[3,3] = cos(a);
finish;
start R13(t,a);
      t = I(5);
      t[1,1] = cos(a);  t[1,3] = sin(a);
      t[3,1] = -sin(a);  t[3,3] = cos(a);
finish;
start R12(t,a);
      t = I(5);
      t[1,1] = cos(a);  t[1,2] = sin(a);
      t[2,1] = -sin(a);  t[2,2] = cos(a);
finish;
start orthmat(t,th1,th2,th3,th4,th5,th6,th7,th8,th9,th10);
      call R45(m1,th1);
      call R35(m2,th2);
      call R25(m3,th3);
      call R15(m4,th4);
      call R34(m5,th5);
      call R24(m6,th6);
      call R14(m7,th7);
      call R23(m8,th8);
      call R13(m9,th9);
      call R12(m10,th10);
      t = m1*m2*m3*m4*m5*m6*m7*m8*m9*m10;
finish;
* Read data;
* stop = number of different clusters in the dataset;
%macro calculate(stop=16);
* Set a matrix to store the output;
  barn = j(&stop,11,-1);
```

```
* Parameters;
  d = 5;
  stop = 3141.6;
  inc = 100;
%do i = 1 %to &stop;
use thesis.bullets2;
    read all where(cluster = &i)
         var{antimony copper arsenic bismuth silver} into X;
close thesis.bullets2;
S1 = x`*x; * Covariance matrix;
call eigen(ls,es,s&i); /* Calculate eigenvalues and eigenvectors */
barn[&i,11] = ls[1]##2*3.141593*&stop;
do th1 = 0 to stop by inc;
 do th2 = 0 to stop by inc;
  do th3 = 0 to stop by inc;
   do th4 = 0 to stop by inc;
    do th5 = 0 to stop by inc;
     do th6 = 0 to stop by inc;
      do th7 = 0 to stop by inc;
       do th8 = 0 to stop by inc;
        do th9 = 0 to stop by inc;
         do th10 = 0 to stop by inc;
         call orthmat(t,th1/1000,th2/1000,th3/1000,th4/1000,th5/1000,
                      th6/1000,th7/1000,th8/1000,th9/1000,th10/1000);
         ts = t*es;    * Rotates the eigenvectors;
         angle = arcos((I(d)#ts)[+,]);
       * Evaluates the target function and stores its value if it is
         smaller than value obtained in the previous iteration;
         dist = ((ls[1:(d-1)]##2)#(angle[1,1:(d-1)])`)[+];
         if dist[1] < barn[&i,11] then do;
            barn[&i,1] = th1/1000; barn[&i,2] = th2/1000;
            barn[&i,3] = th3/1000; barn[&i,4] = th4/1000;
            barn[&i,5] = th5/1000; barn[&i,6] = th6/1000;
            barn[&i,7] = th7/1000; barn[&i,8] = th8/1000;
            barn[&i,9] = th9/1000; barn[&i,10] = th10/1000;
            barn[&i,11] = dist;
         end;
         end; * ends th10-loop;
        end; * ends th9-loop;
       end; * ends th8-loop;
      end; * ends th7-loop;
     end; * ends th6-loop;
    end; * ends th5-loop;
   end; * ends th4-loop;
```

```
  end; * ends th3-loop;
 end; * ends th2-loop;
end; * ends th1-loop;
%end;
%mend;
%calculate;
* Print results;
title 'Rotations on R5';
print barn;
create thesis.barn from barn;
       append from barn;
close thesis.barn;
quit;
/*************************************************************\
|*************************************************************|
| Define module to simulate pseudo-random matrices from       |
| the matrix Langevin distribution with parameter F.          |
|*************************************************************|
|*************************************************************|
|Parameters: F = an n*m matrix of parameters                   |
|            showmat = an scalar. A value of 1 shows the      |
|                      pseudo-matrix generated; 2 shows       |
|                      all information generated by the       |
|                      algorithm; 3 shows how many steps      |
|                      were needed to generate the desired    |
|                      matrix; any other value suppresses      |
|                      the output.                            |
|            method = "C" Chikuse's                           |
|                     "GEA" Generalized Euler angles.         |
|                     Method controls how the pseudo-random   |
|                     uniform matrices are generated.         |
|            restrict = 1 the generated matrix is a           |
|                          member of SO(n). Any other value   |
|                          generates a matrix on V_{m,n}.     |
|      seed_L = seed to initialize the rannor function.       |
|      seed_u = seed to initialize the ranuni function.       |
|     NOTE THAT ALL THOSE PARAMETERS ARE GLOBAL               |
|    (i.e. each parameter has to be set to a                  |
|     value before invoking the module.)                      |
\*************************************************************/
proc iml;
reset fuzz storage = thesis.simulation;
* Chikuse's method to generate a pseudo-random uniform matrix on
  V_{m,n};
```

```
start Chikuse;
        n0= nrow(F);
        m0 = ncol(F);
        tmp1 = j(n0,m0,.);
        call randgen(tmp1,'normal');
        temp0 = root(temp1'*temp1);
        if det(temp0) ^= 0 then do;
           temp2 = temp1*inv(temp0);
           temp21 = trace(F'*temp2-diag(delta));
        end;
        else temp21 = 0;
finish Chikuse;
* Generalized Euler angles method to generate a pseudo-random
  uniform matrix on V_{m,n};
start GEAngles;
length = nnn*(nnn-1)/2;
angles0 = j(length,1,.);
call randseed(seed_L,p);
call randgen(angles0,'uniform');
call sort(angles0,{1},{1});
angles = j(length,1,2*3.1415926535897)#(angles0);
angles[length] = (3.1415926535897)#(angles0[length]);
*T1;
initial = I(2);
if restrict ^= 1 then do;
   bb = ranbin(9,1,.5);
   initial[1,1]=(-1#bb + (1-bb));
end;
*T2; r12 = I(2);
*R12;
r12[1,1] = cos(angles[1]); r12[2,2] = cos(angles[1]);
r12[2,1] = sin(angles[1]); r12[1,2] = -sin(angles[1]);
T2=r12*initial;
*T3;
smallt3=I(3);
smallt3[1:2,1:2] = T2;
r23 = I(3); r13 = I(3);
*R23;
r23[2,2] = cos(angles[2]); r23[3,3] = cos(angles[2]);
r23[3,2] = sin(angles[2]); r23[2,3] = -sin(angles[2]);
*R13;
r13[1,1] = cos(angles[3]); r13[3,3] = cos(angles[3]);
r13[3,1] = sin(angles[3]); r13[1,3] = -sin(angles[3]);
T3 = r23*r13*smallt3;
```

```
*T4;
smallt4 = I(4);
smallt4[1:3,1:3] = T3;
r34 = I(4); r24 = I(4); r14 = I(4);
*R34;
r34[3,3] = cos(angles[4]); r34[4,4] = cos(angles[4]);
r34[4,3] = sin(angles[4]); r34[3,4] = -sin(angles[4]);
*R24;
r24[2,2] = cos(angles[5]); r24[4,4] = cos(angles[5]);
r24[4,2] = sin(angles[5]); r24[2,4] = -sin(angles[5]);
*R14;
r14[1,1] = cos(angles[6]); r14[4,4] = cos(angles[6]);
r14[4,1] = sin(angles[6]); r14[1,4] = -sin(angles[6]);
T4 = r34*r24*r14*smallt4;
if nnn = 4 then temp2 = t4;
else if nnn = 5 then do;
*T5;
smallt5 = I(5);
smallt5[1:4,1:4]=T4;
r45 = I(5); r35 = I(5); r25 = I(5); r15 = I(5);
*R45;
r45[4,4] = cos(angles[7]); r45[5,5] = cos(angles[7]);
r45[5,4] = sin(angles[7]); r45[4,5] = -sin(angles[7]);
*R35;
r35[3,3] = cos(angles[8]); r35[5,5] = cos(angles[8]);
r35[5,3] = sin(angles[8]); r35[3,5] = -sin(angles[8]);
*R25;
r25[2,2] = cos(angles[9]); r25[5,5] = cos(angles[9]);
r25[5,2] = sin(angles[9]); r25[2,5] = -sin(angles[9]);
*R15;
r15[1,1] = cos(angles[10]); r15[5,5] = cos(angles[10]);
r15[5,1] = sin(angles[10]); r15[1,5] = -sin(angles[10]);
temp2 = r45*r35*r25*r15*smallt5;
end;
angles0 = angles`#(180/3.1415926535897);
finish GEAngles;
start rLangevin;
k = 1; * k is to keep track of how many iterations are
         needed to generate a random matrix;
stop = 0;
call svd(null1,delta,null2,F);
  do until (stop = 1);
     u = ranuni(seed_u);
     * Select method to generate pseudo-random uniform matrices;
```

```
     if method = "C" then run Chikuse;
     else do;
          run GEAngles;
  temp20 = temp2;
      temp2 = temp2[,1:(ncol(F))];
          temp21 = trace(F'*temp2-diag(delta));
     end;
     * Simple rejection method;
     temp1 = exp(temp21);
     if method = "C" then do;
        if restrict = 1 then do;
           temp3 = det(temp2);
           if (u < temp1 & temp3 = 1) then stop = 1;
        end;
        else if (u < temp1) then stop = 1;
 end;
     else if (u < temp1) then stop = 1;
     k = k + 1;
  end;
     if showmat = 1 then do;
        print temp2;
     end;
 if showmat = 2 then do;
        temp3 = det(temp20);
        print p k u temp1 temp3 temp2;
     end;
 else if showmat = 3 then do;
print k;
 end;
finish;
store module = (rLangevin GEAngles Chikuse);
quit;
* Gibbs sampler;
* The macro parameter factor is needed only to do sensitivity analysis;
%macro gibbsMLan(nchains,ndraws,factor=1);
%do chain = 1 %to &nchains;
proc iml;
reset fuzz storage = thesis.simulation;
************************************************************
* Set global parameters
************************************************************;
seed_L = 9*&chain;
seed_u = 8*&chain;
method = "GEA";
```

```
showmat = 0;
restrict = 0;
p = 0;
*************************************************************
* Load module rLangevin and associated modules
*************************************************************;
load module = (rLangevin Chikuse GEAngles);
load barn;
n = nrow(barn);
m = ncol(barn);
*************************************************************
* Set the matrix of eigenvalues of F
*************************************************************;
load lambdas;
Dphi = diag(&factor*lambdas);
*************************************************************
* Set length of chain and matrix to store draws
*************************************************************;
ndraws = &ndraws;
collect_G = j(ndraws,n*m,.);
collect_D = j(ndraws,m*m,.);
collect_AG = j(ndraws,n*(n-1)/2,.);
collect_AD = j(ndraws,m*(m-1)/2,.);
*************************************************************
* Set initial value for Delta
*************************************************************;
nnn = m;
F = I(m);
run rLangevin;
tmp2 = temp2;
*************************************************************
* Gibbs sampler
*************************************************************;
do pp = 1 to ndraws;
* Draw Gamma from its conditional distribution;
F_G = (Dphi*tmp2*barn`)`;
F = F_G;
nnn = n; restrict = 1;
run rLangevin;
tmp1 = temp2;
collect_G[pp,] = shape(tmp1,1,n*m);
if method = "GEA" then collect_AG[pp,] = angles0;
* Draw Delta from its conditional distribution;
F_D = Dphi*tmp1`*barn;
```

```
F = F_D;
nnn = m; restrict = 0;
run rLangevin;
tmp2 = temp2;
collect_D[pp,] = shape(tmp2,1,m*m);
if method = "GEA" then collect_AD[pp,] = angles0;
end;
create Delta from collect_D;
append from collect_D;close Delta;
create Gamma from collect_G;
append from collect_G;close Gamma;
create Angles_D from collect_AD;
append from collect_AD;close Angles_D;
create Angles_G from collect_AG;
append from collect_AG;close Angles_G;
quit;
data thesis.Gamma_c&chain;
     set gamma;
 iteration = _N_;
data thesis.Delta_c&chain;
     set Delta;
 iteration = _N_;
data thesis.AnglesG_c&chain;
     set Angles_G;
 iteration = _N_;
data thesis.AnglesD_c&chain;
     set Angles_D;
 iteration = _N_;
run;
%end;
%mend;
%gibbsMLan(4,100000)
```

APPENDIX C Gibbs sampler diagnostics

Diagnostics for Δ

Table C.1 shows the Gelman and Rubin diagnostic ([18]) for the components of Δ. Figure C.1 and Figure C.2 show the plots of the shrink factor. Autocorrelations are shown on Figures C.3 through C.6; only plots for the first chain are shown as plots for the other three chains are very similar. Figures C.7 through C.10 show the trace of the four chains for the last 500 iterations of the Gibbs sampler. All plots are organized by row of Δ. Figure C.11 shows the cross-correlations of the parameters by chain. Tables C.2 through C.5 show the Raftery and Lewis diagnostic ([36]).

Table C.1 Gelman and Rubin diagnostic: Δ.

Estimand	Point est.	97.5% quantile	Estimand	Point est.	97.5% quantile
$[1,1]$	1	1	$[3,1]$	1	1
$[1,2]$	1	1	$[3,2]$	1	1
$[1,3]$	1	1	$[3,3]$	1	1
$[1,4]$	1	1	$[3,4]$	1	1
$[2,1]$	1	1	$[4,1]$	1	1
$[2,2]$	1	1	$[4,2]$	1	1
$[2,3]$	1	1	$[4,3]$	1	1
$[2,4]$	1	1	$[4,4]$	1	1

$[i,j]$ represents the element of the ith row and jth column

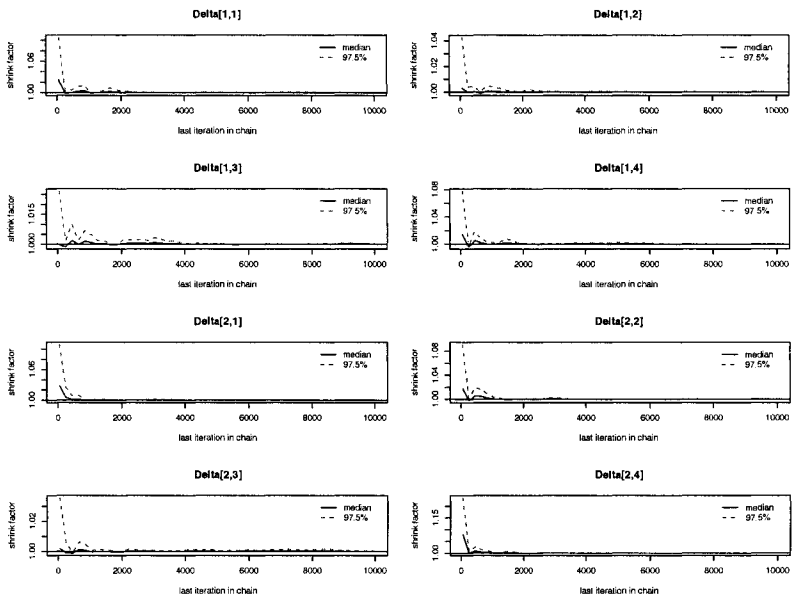

Figure C.1 Shrink factor plots of the components of the first and second rows of Δ.

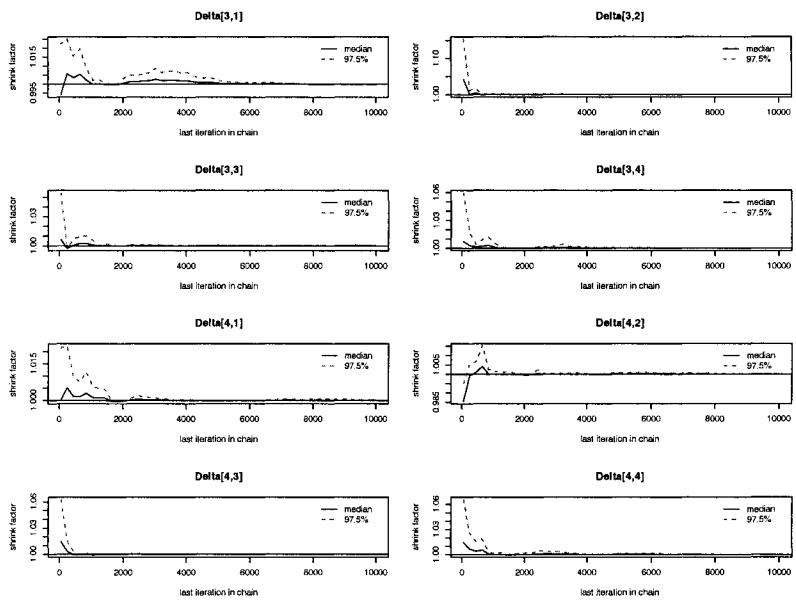

Figure C.2 Shrink factor plots of the components of the third and fourth rows of Δ.

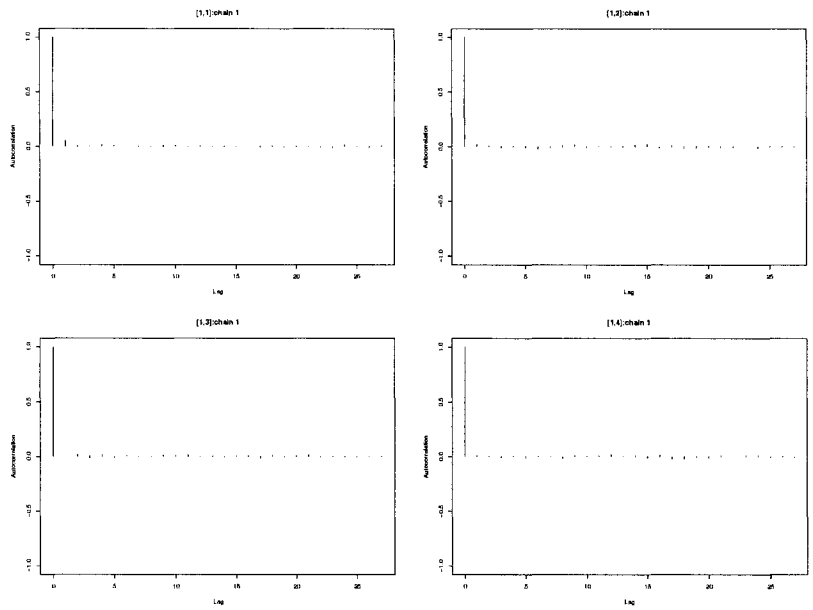

Figure C.3 Autocorrelation for chain 1 of the components of the first row of Δ.

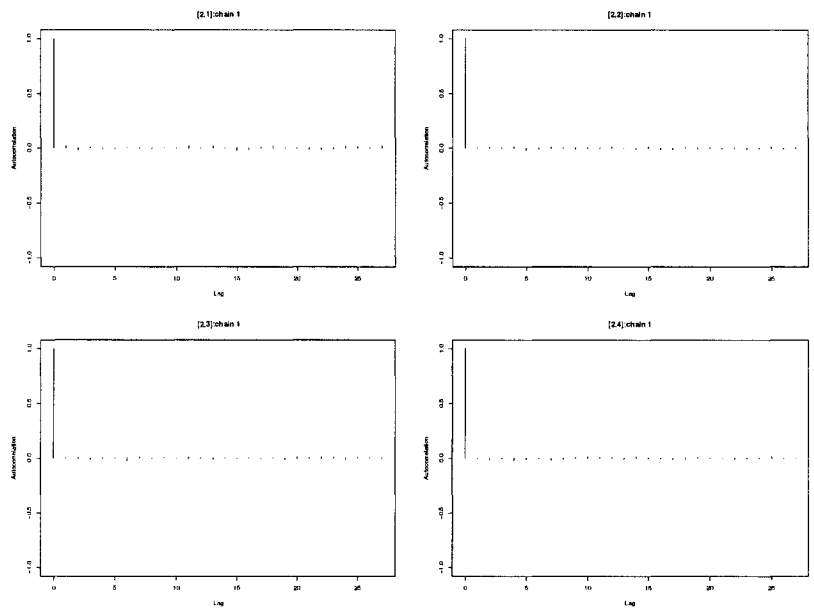

Figure C.4 Autocorrelation for chain 1 of the components of the second row of Δ.

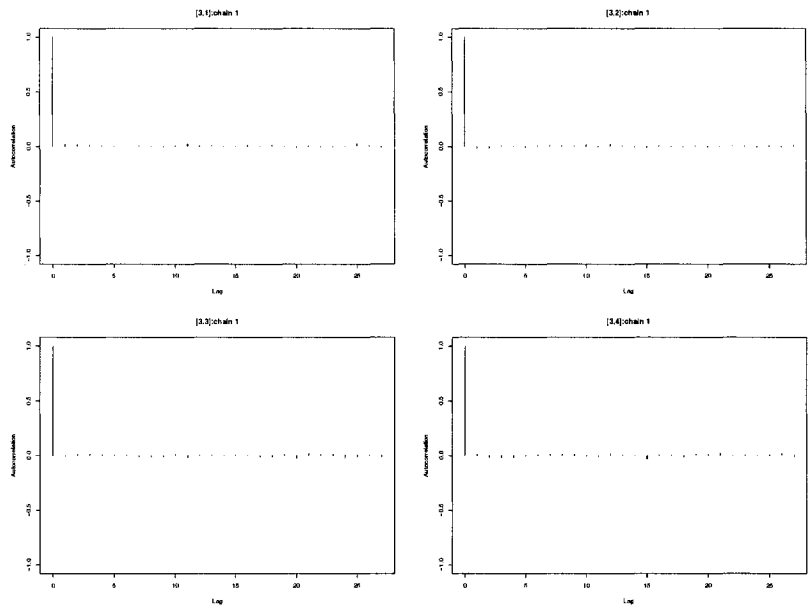

Figure C.5 Autocorrelation for chain 1 of the components of the third row of Δ.

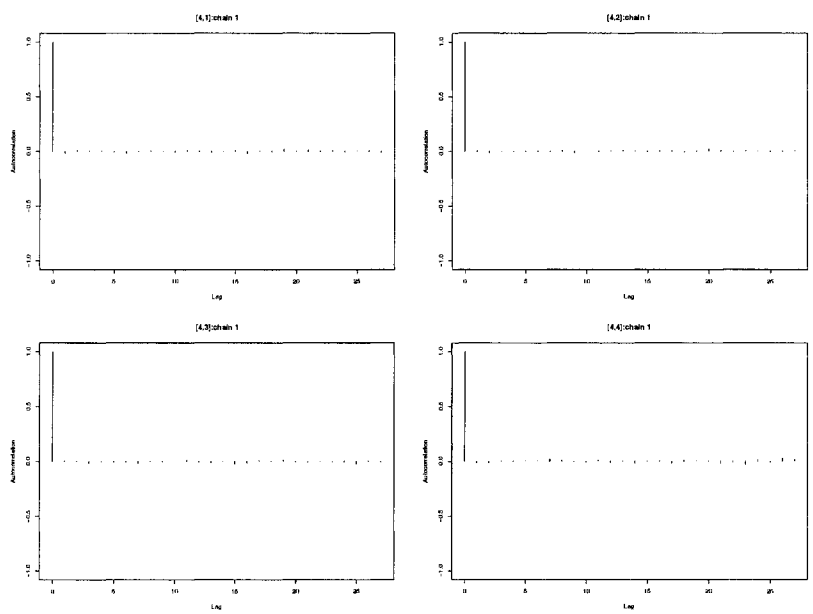

Figure C.6 Autocorrelation for chain 1 of the components of the fourth row of Δ.

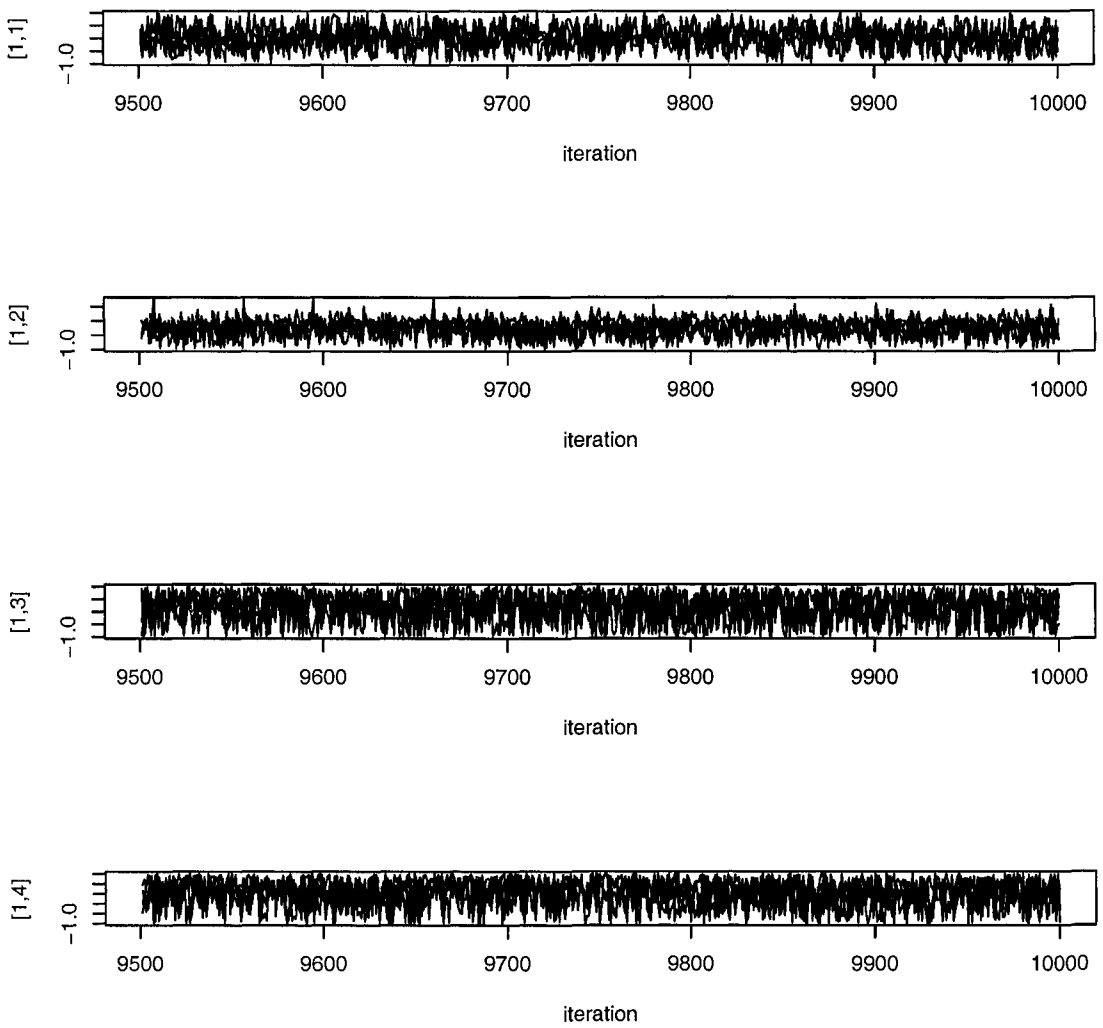

Figure C.7 Trace of the last 500 iterations of four simulated sequences of the components of the first row of Δ.

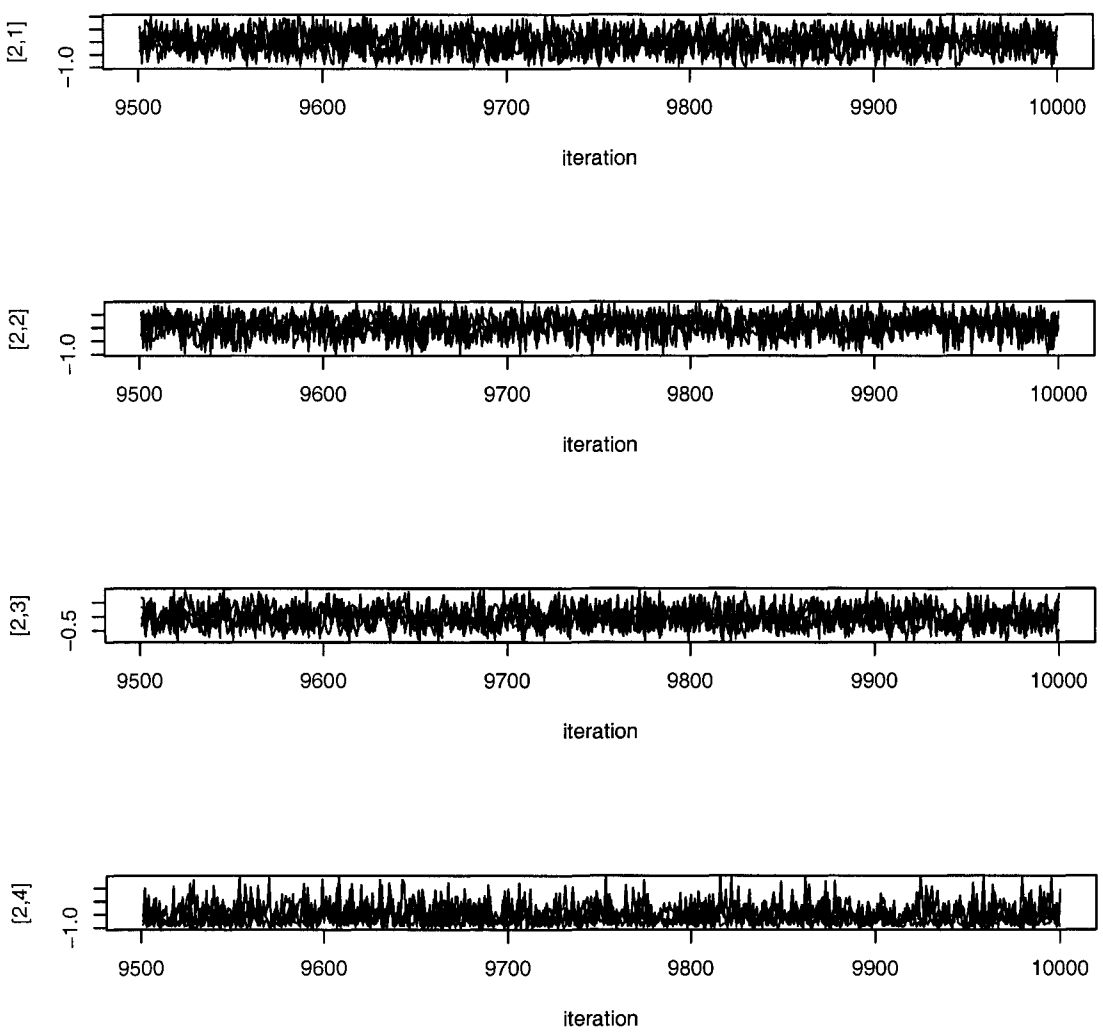

Figure C.8 Trace of the last 500 iterations of four simulated sequences of the components of the second row of Δ.

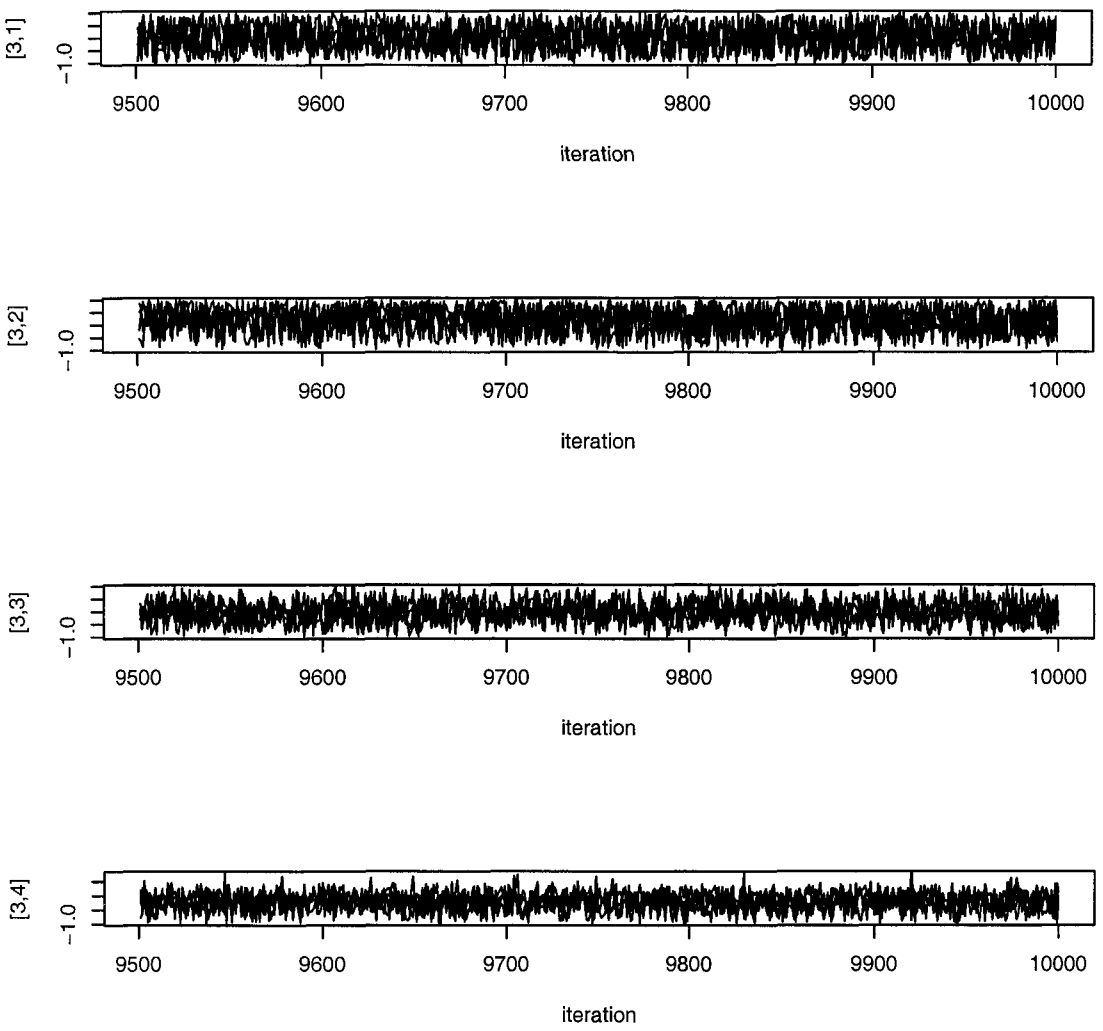

Figure C.9 Trace of the last 500 iterations of four simulated sequences of the components of the third row of Δ.

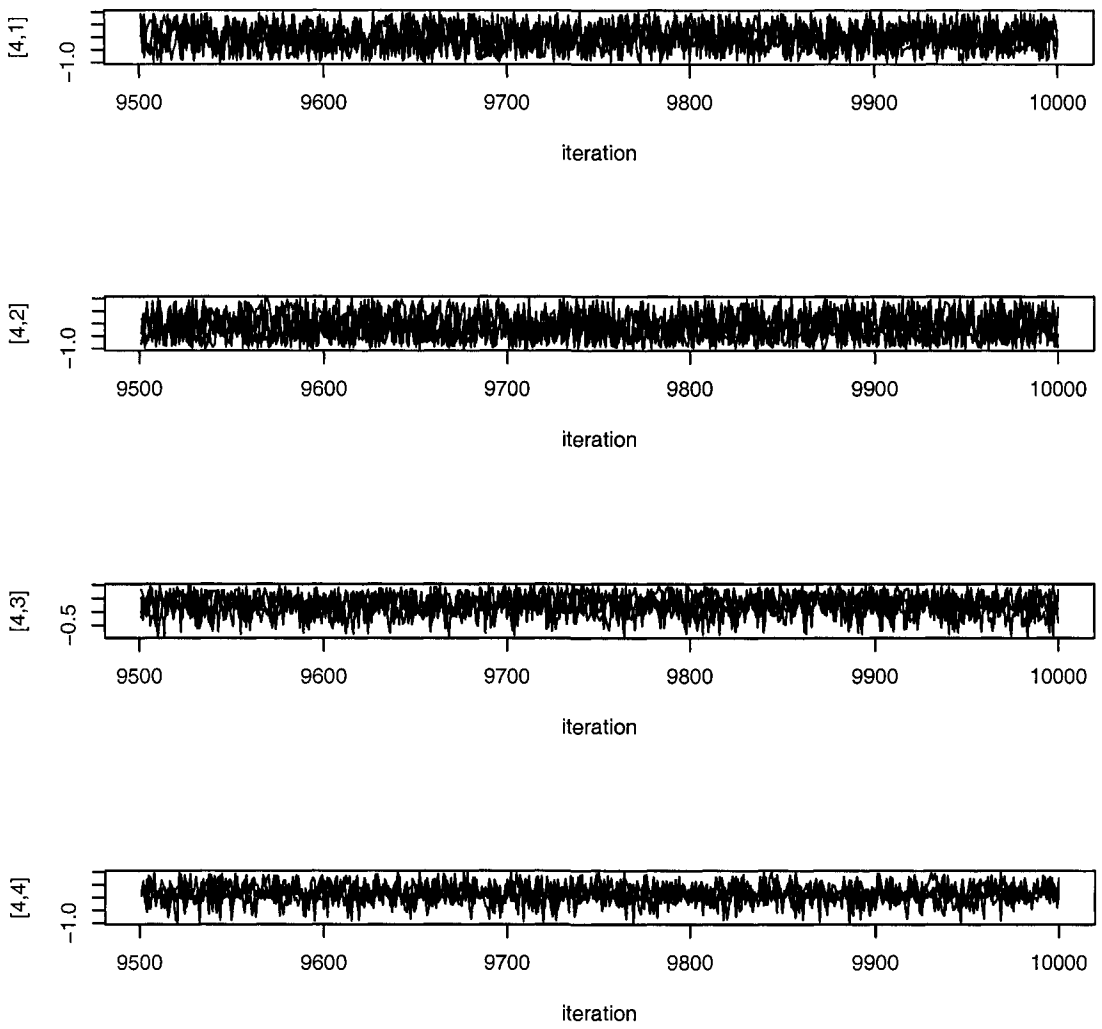

Figure C.10 Trace of the last 500 iterations of four simulated sequences of the components of the fourth row of Δ.

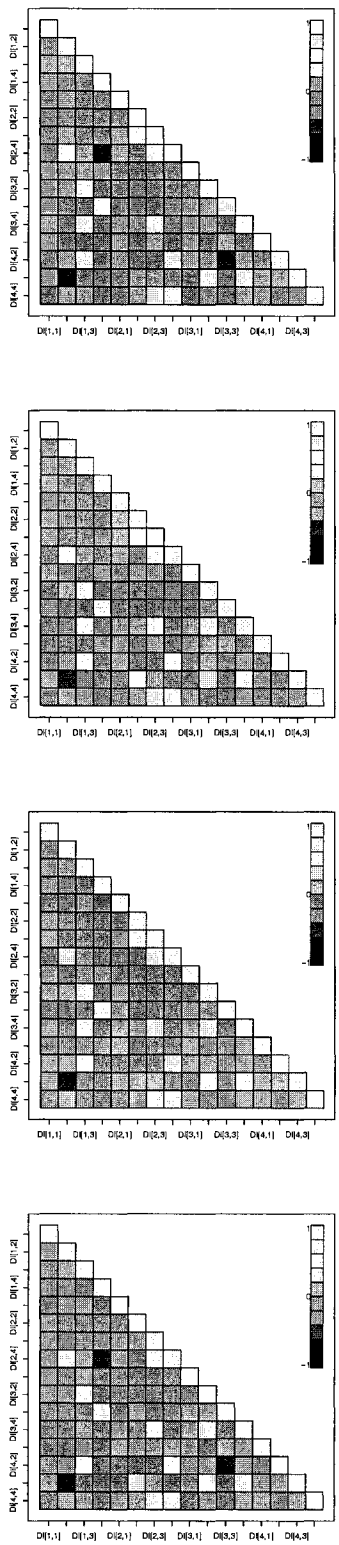

Figure C.11 Crosscorrelations among parameters by chain (chains are ordered from top to bottom.)

Table C.2 Δ: Raftery and Lewis diagnostic (chain 1).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3802	3746	1.010
$[1,2]$	2	3741	3746	0.999
$[1,3]$	2	3897	3746	1.040
$[1,4]$	2	3710	3746	0.990
$[2,1]$	2	3865	3746	1.030
$[2,2]$	2	3710	3746	0.990
$[2,3]$	2	3865	3746	1.030
$[2,4]$	2	3710	3746	0.990
$[3,1]$	2	3741	3746	0.999
$[3,2]$	2	3741	3746	0.999
$[3,3]$	2	3802	3746	1.010
$[3,4]$	2	3834	3746	1.020
$[4,1]$	2	3834	3746	1.020
$[4,2]$	2	3802	3746	1.010
$[4,3]$	2	3771	3746	1.010
$[4,4]$	2	3680	3746	0.982

$[i,j]$ represents the element of the ith row and jth column

Table C.3 Δ: Raftery and Lewis diagnostic (chain 2).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3710	3746	0.990
$[1,2]$	2	3771	3746	1.010
$[1,3]$	2	3802	3746	1.010
$[1,4]$	2	3741	3746	0.999
$[2,1]$	2	3834	3746	1.020
$[2,2]$	2	3680	3746	0.982
$[2,3]$	2	3929	3746	1.050
$[2,4]$	2	3834	3746	1.020
$[3,1]$	2	3771	3746	1.010
$[3,2]$	2	3897	3746	1.040
$[3,3]$	2	3771	3746	1.010
$[3,4]$	2	3680	3746	0.982
$[4,1]$	2	3710	3746	0.990
$[4,2]$	2	3819	3746	1.020
$[4,3]$	2	3771	3746	1.010
$[4,4]$	2	3741	3746	0.999

$[i,j]$ represents the element of the ith row and jth column

Table C.4 Δ: Raftery and Lewis diagnostic (chain 3).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3865	3746	1.030
$[1,2]$	2	3741	3746	0.999
$[1,3]$	2	3590	3746	0.958
$[1,4]$	2	3771	3746	1.010
$[2,1]$	2	3710	3746	0.990
$[2,2]$	2	3710	3746	0.990
$[2,3]$	2	3834	3746	1.020
$[2,4]$	2	3834	3746	1.020
$[3,1]$	2	3741	3746	0.999
$[3,2]$	2	3802	3746	1.010
$[3,3]$	2	3741	3746	0.999
$[3,4]$	2	3802	3746	1.010
$[4,1]$	2	3802	3746	1.010
$[4,2]$	2	3650	3746	0.974
$[4,3]$	2	3802	3746	1.010
$[4,4]$	2	3710	3746	0.990

$[i,j]$ represents the element of the ith row and jth column

Table C.5 Δ: Raftery and Lewis diagnostic (chain 4).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3771	3746	1.010
$[1,2]$	2	3650	3746	0.974
$[1,3]$	2	3754	3746	1.000
$[1,4]$	2	3802	3746	1.010
$[2,1]$	2	3865	3746	1.030
$[2,2]$	2	3680	3746	0.982
$[2,3]$	2	3802	3746	1.010
$[2,4]$	2	3710	3746	0.990
$[3,1]$	2	3741	3746	0.999
$[3,2]$	2	3741	3746	0.999
$[3,3]$	2	3741	3746	0.999
$[3,4]$	2	3771	3746	1.010
$[4,1]$	2	3897	3746	1.040
$[4,2]$	2	3650	3746	0.974
$[4,3]$	2	3696	3746	0.987
$[4,4]$	2	3771	3746	1.010

$[i,j]$ represents the element of the ith row and jth column

Diagnostics for Γ

Figure C.12 and C.13 show the plots of Gelman and Rubin shrink factor ([18]) for the components of Γ, numerical values are shown on Table C.6. Autocorrelations are shown on Figures C.14 through C.17; only plots for the first chain are shown as plots for the other three chains are very similar. Figures C.19 through C.23 show the trace of the four chains for the last 500 iterations of the Gibbs sampler. All plots are organized by row of Γ. Figure C.24 shows the cross-correlations of the parameters by chain. Tables C.7 through C.10 show the Raftery and Lewis diagnostic ([36]).

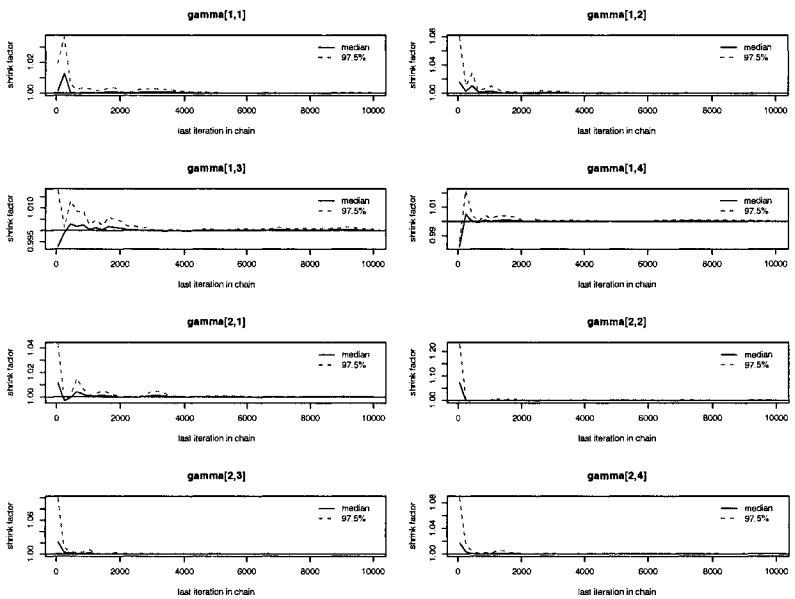

Figure C.12 Shrink factor plots of the components of the first and second rows of Γ.

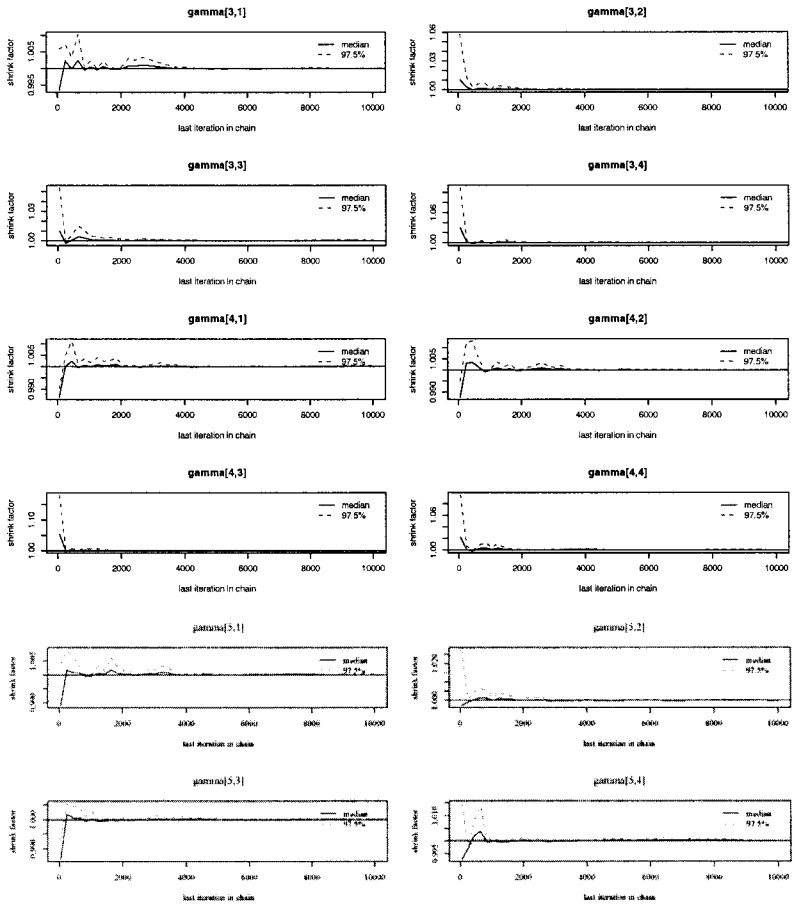

Figure C.13 Shrink factor plots of the components of the third, fourth and fifth rows of Γ.

Table C.6 Γ: Gelman and Rubin diagnostic.

Estimand	Point est.	97.5% quantile	Estimand	Point est.	97.5% quantile
$[1,1]$	1	1	$[3,3]$	1	1
$[1,2]$	1	1	$[3,4]$	1	1
$[1,3]$	1	1	$[4,1]$	1	1
$[1,4]$	1	1	$[4,2]$	1	1
$[2,1]$	1	1	$[4,3]$	1	1
$[2,2]$	1	1	$[4,4]$	1	1
$[2,3]$	1	1	$[5,1]$	1	1
$[2,4]$	1	1	$[5,2]$	1	1
$[3,1]$	1	1	$[5,3]$	1	1
$[3,2]$	1	1	$[5,4]$	1	1

$[i,j]$ represents the element of the ith row and jth column

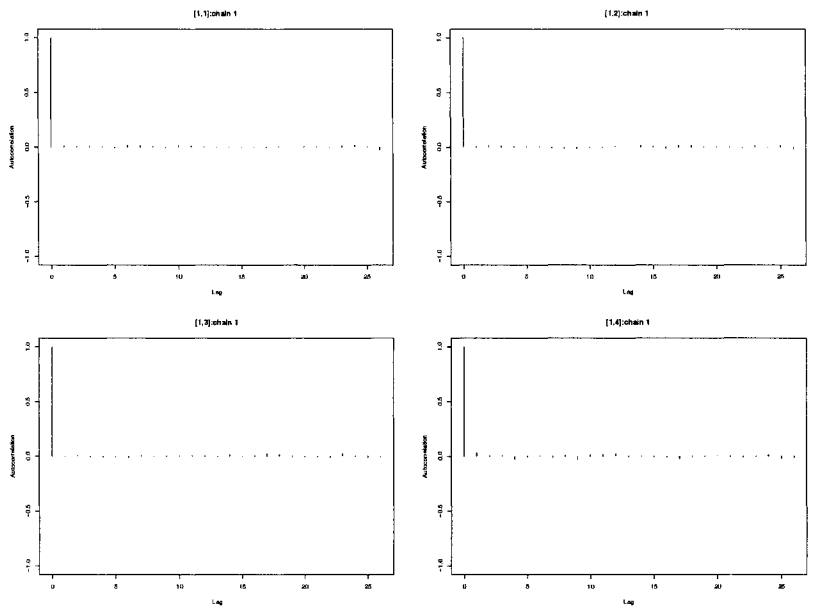

Figure C.14 Autocorrelation for chain 1 of the components of the first row of Γ.

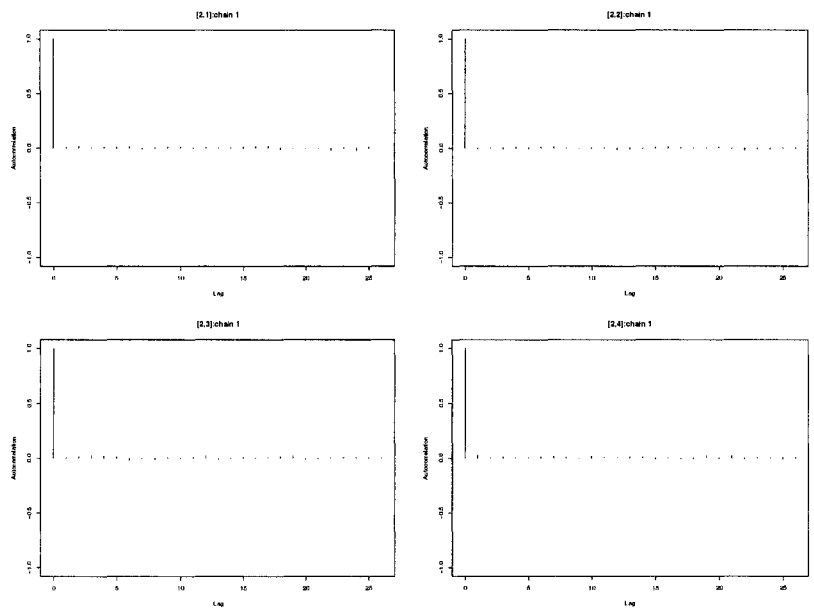

Figure C.15 Autocorrelation for chain 1 of the components of the second row of Γ.

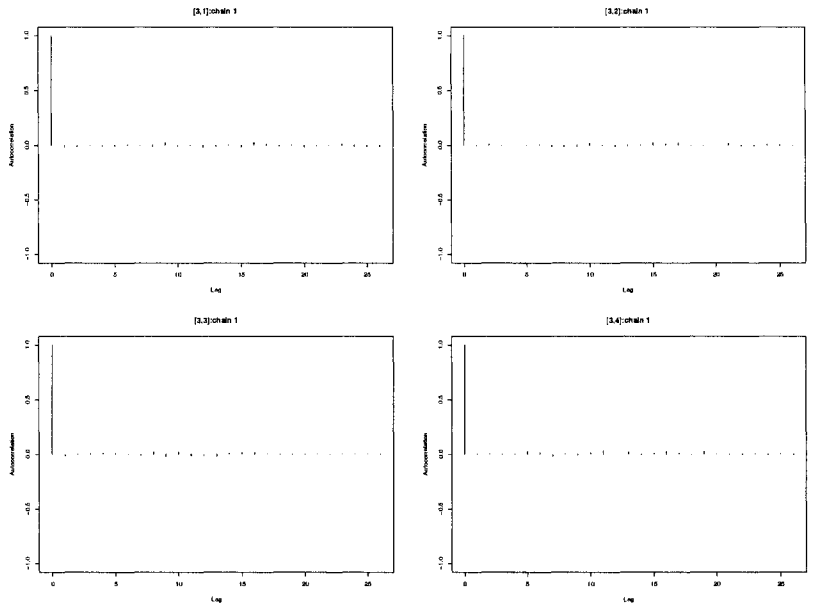

Figure C.16 Autocorrelation for chain 1 of the components of the third row of Γ.

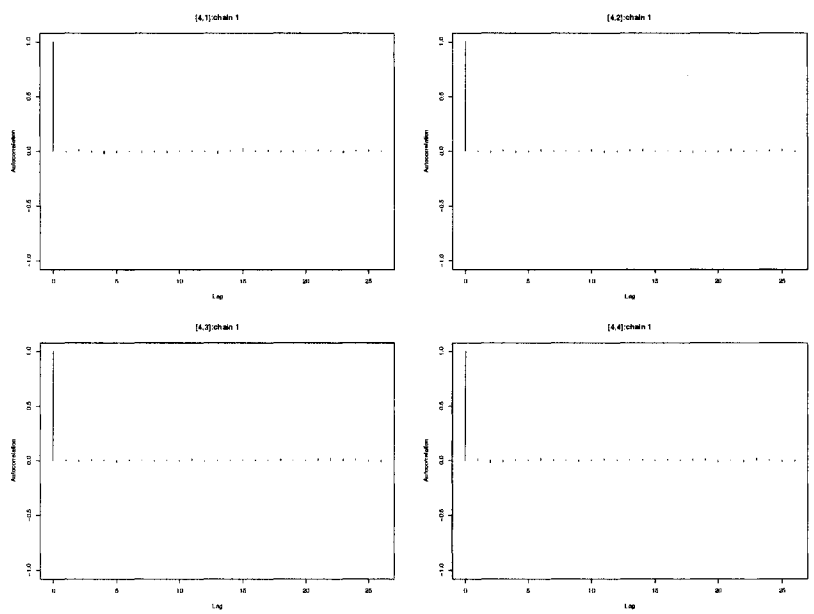

Figure C.17 Autocorrelation for chain 1 of the components of the fourth row of Γ.

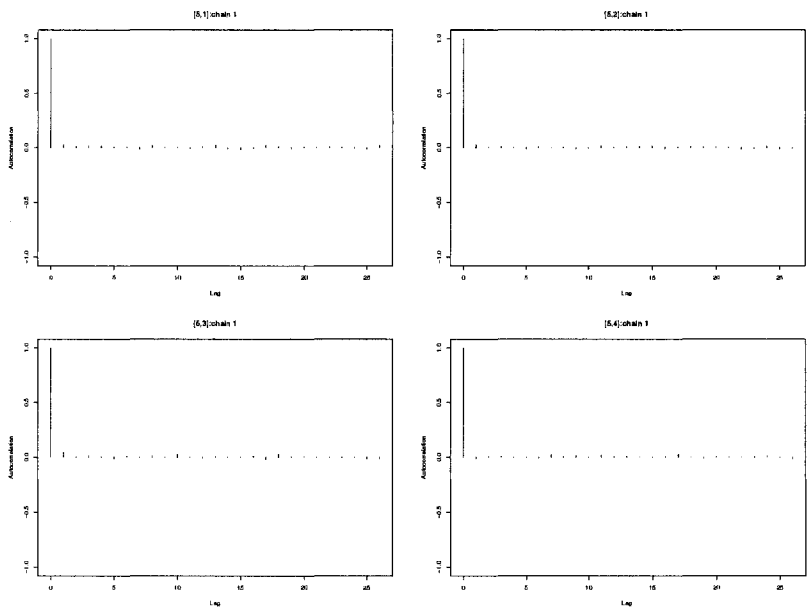

Figure C.18 Autocorrelation for chain 1 of the components of the fifth row of Γ.

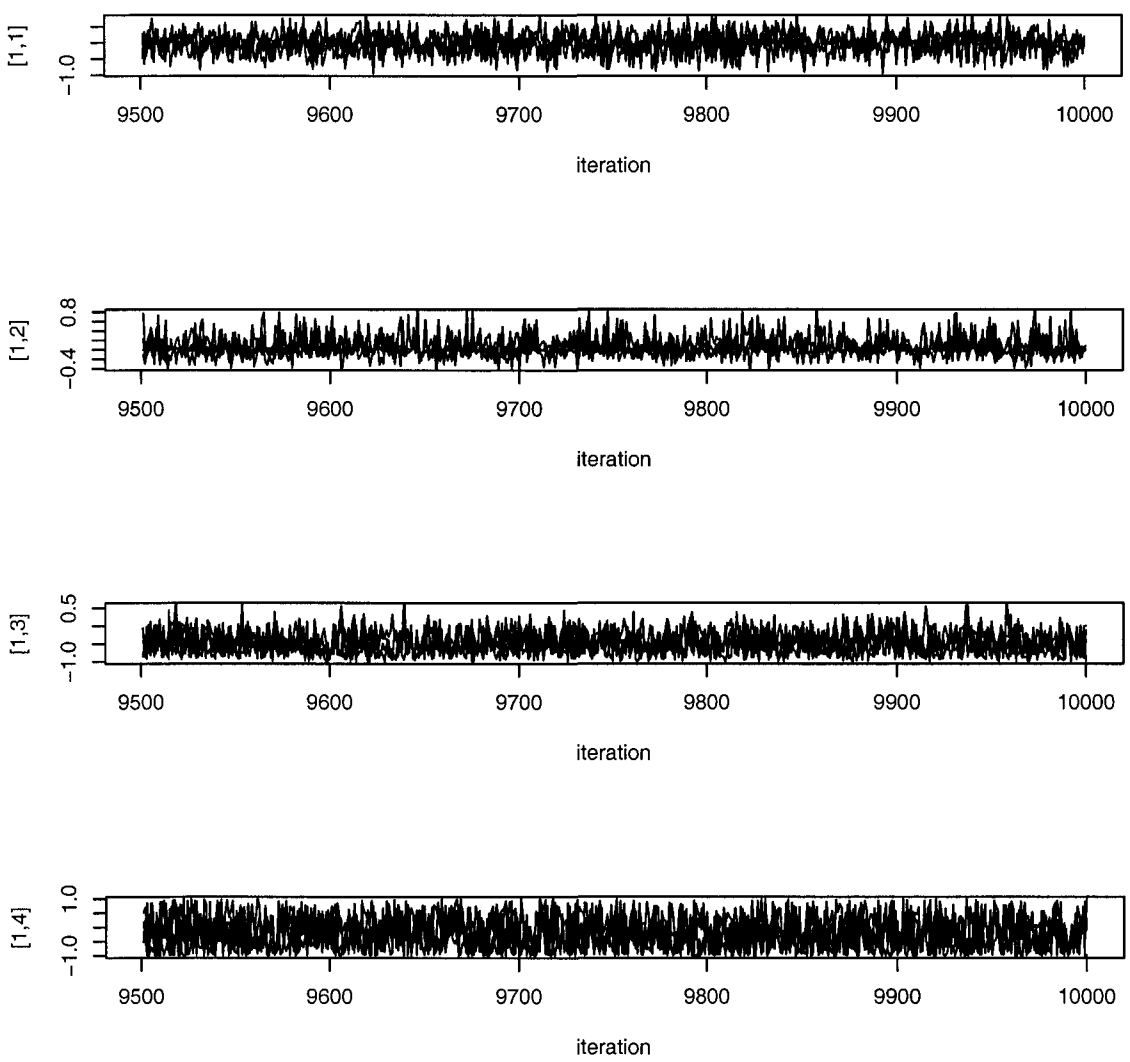

Figure C.19 Trace of the last 500 iterations of four simulated sequences of the components of the first row of Γ.

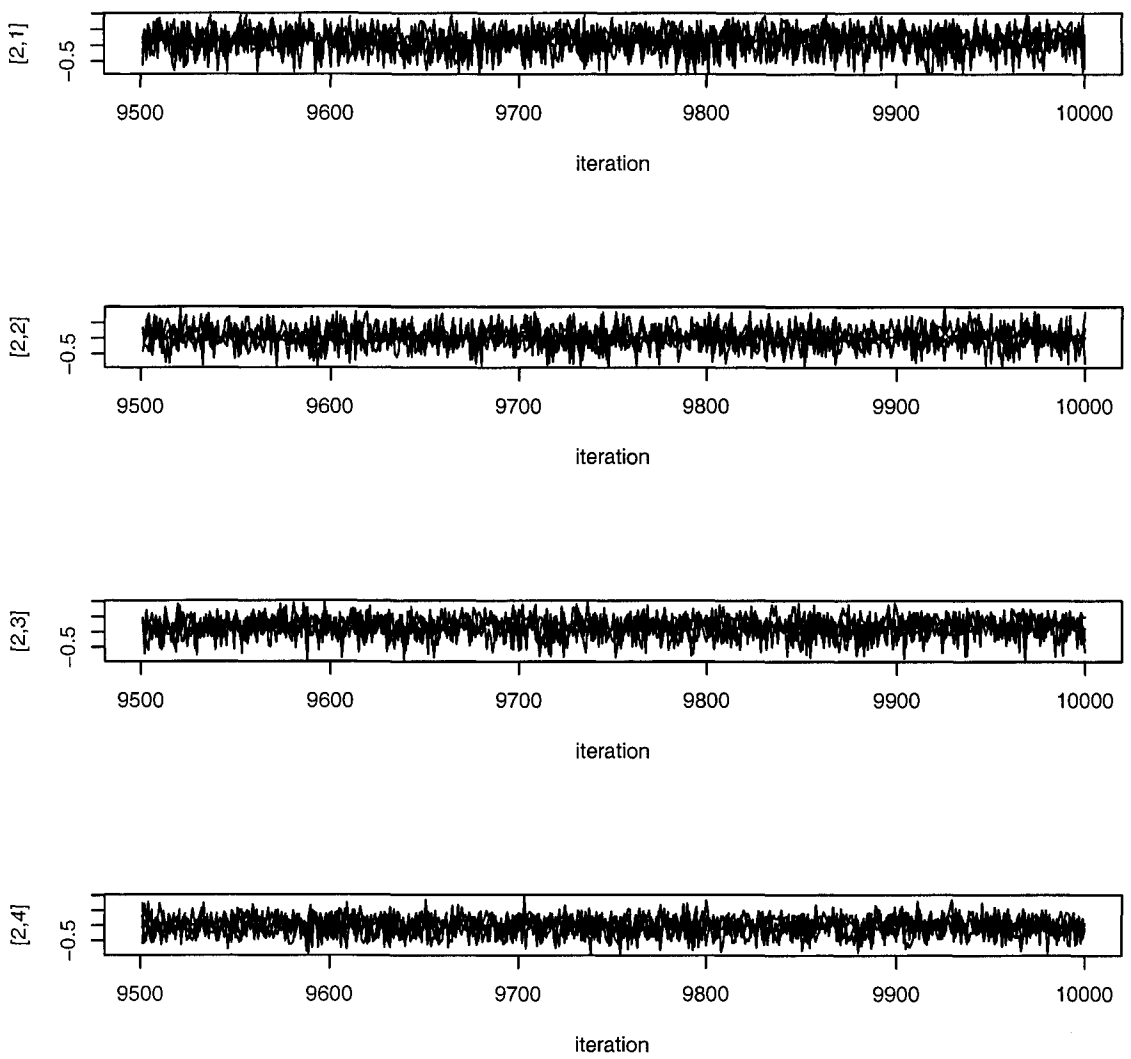

Figure C.20 Trace of the last 500 iterations of four simulated sequences of the components of the second row of Γ.

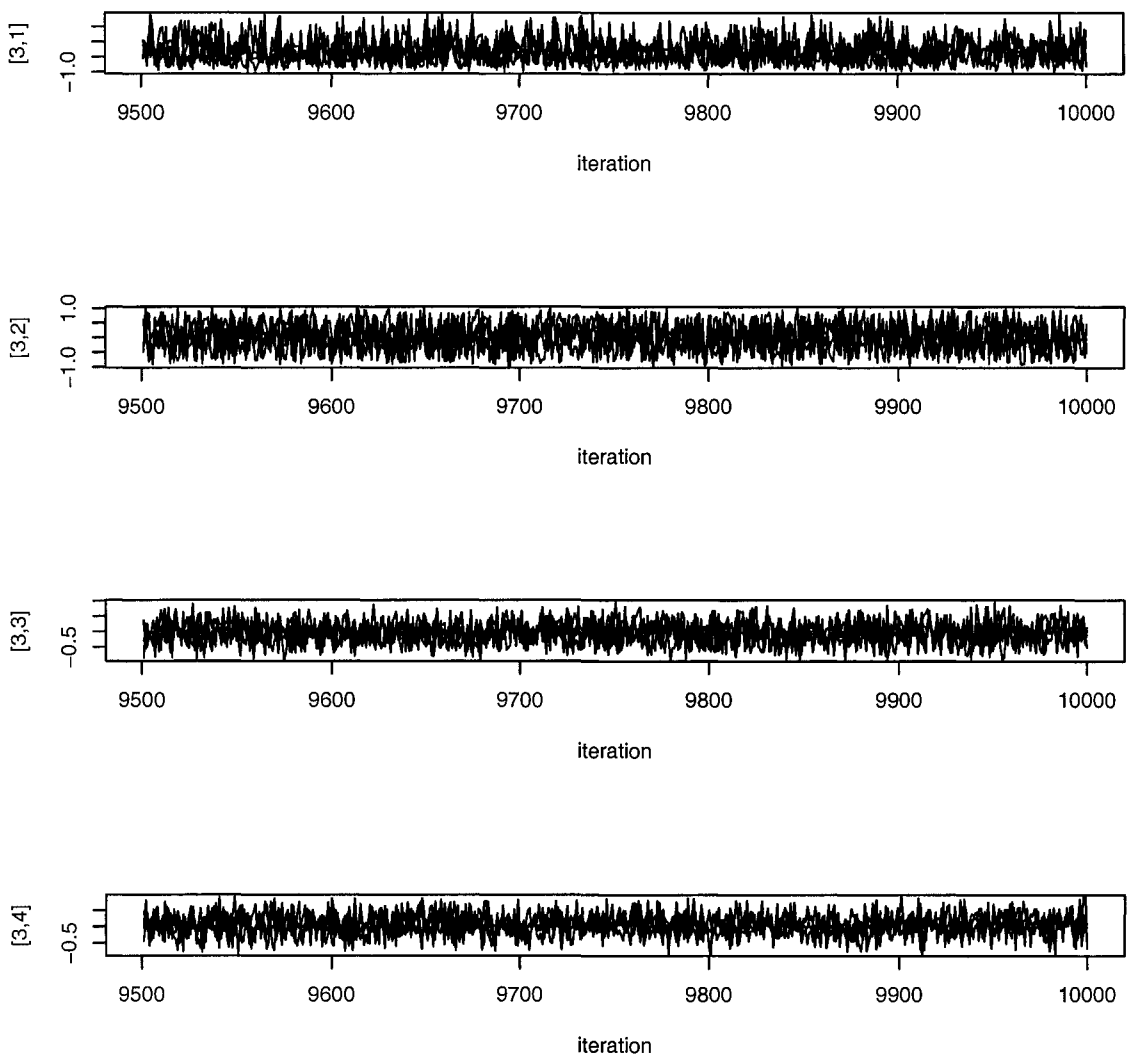

Figure C.21 Trace of the last 500 iterations of four simulated sequences of the components of the third row of Γ.

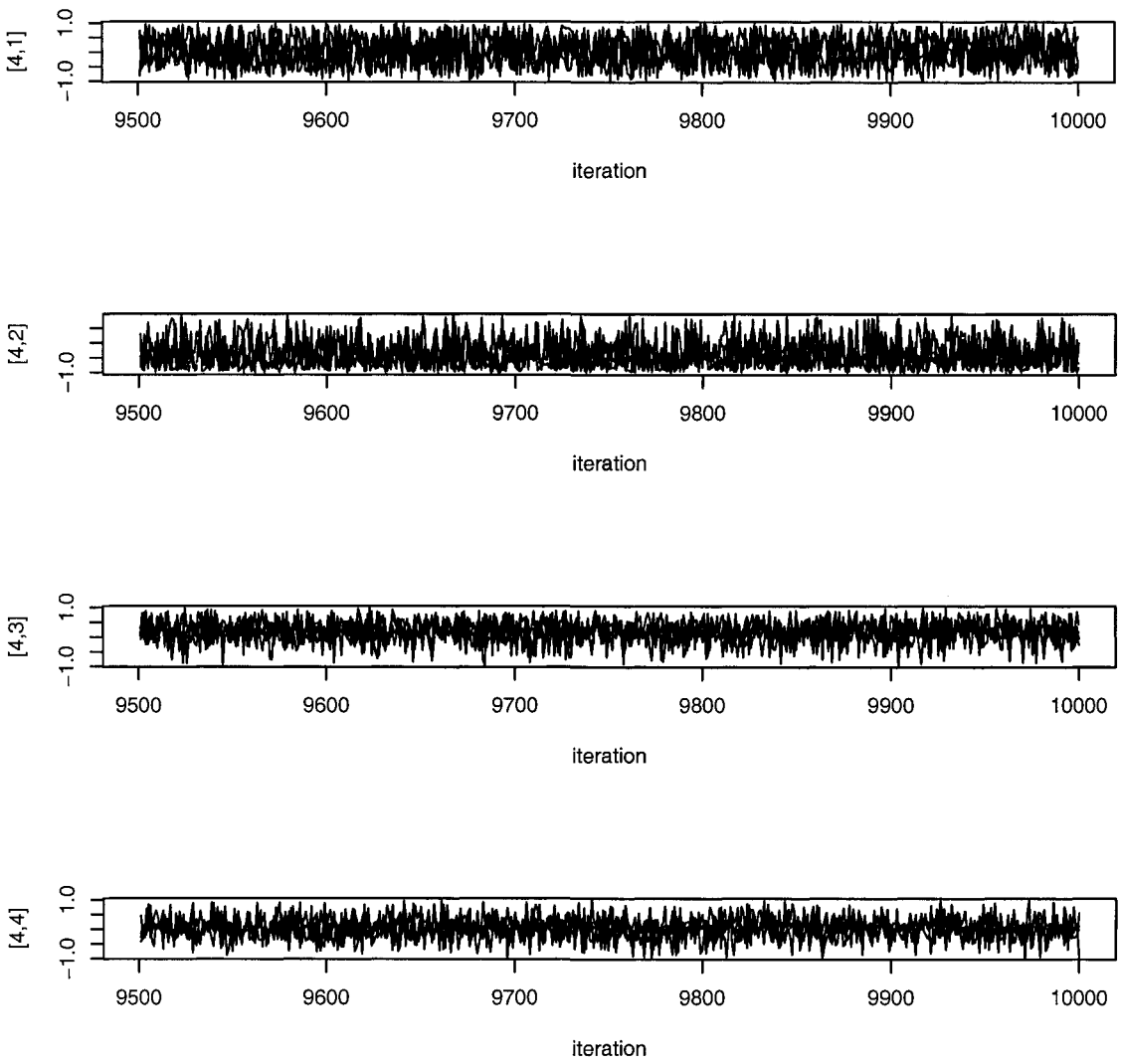

Figure C.22 Trace of the last 500 iterations of four simulated sequences of the components of the fourth row of Γ.

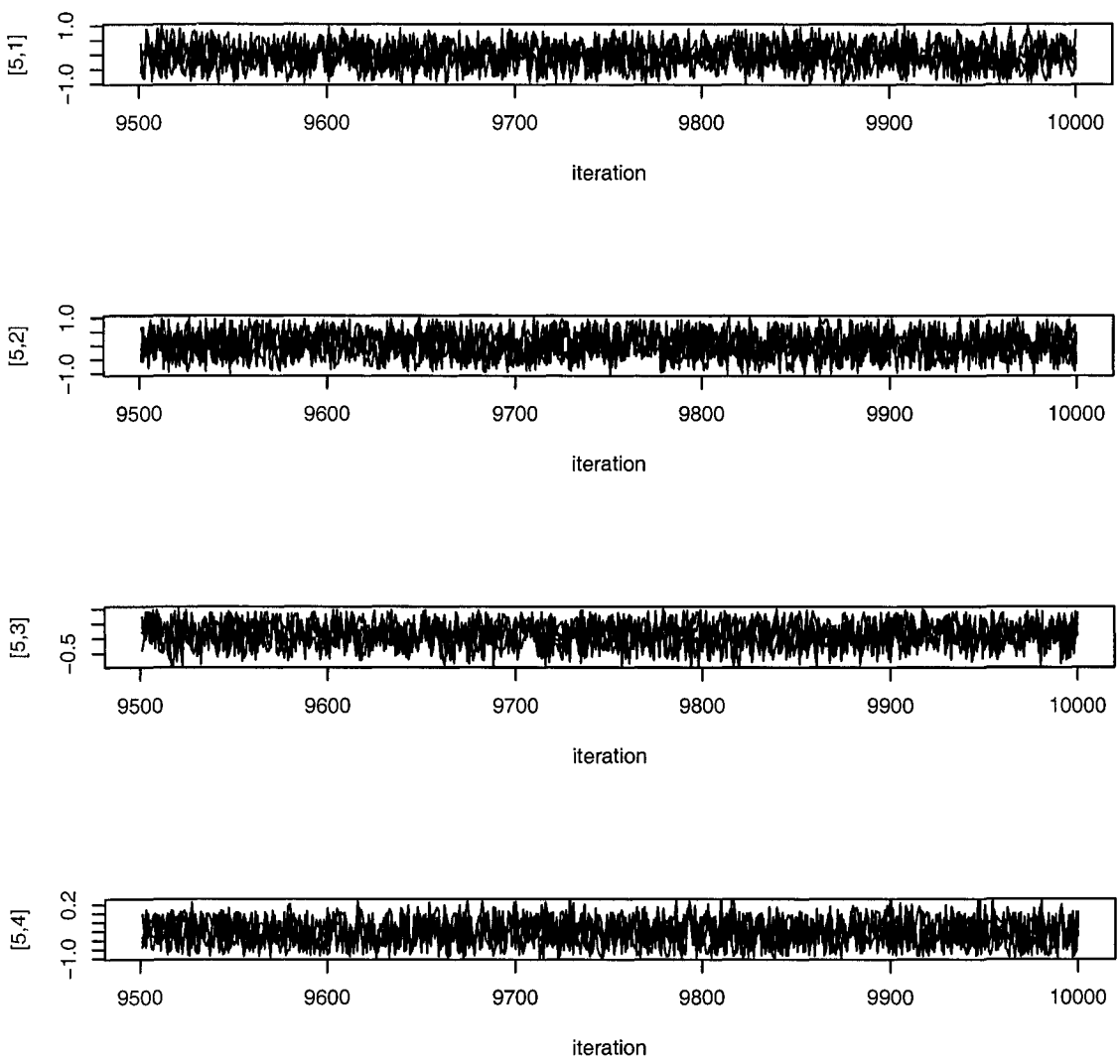

Figure C.23 Trace of the last 500 iterations of four simulated sequences of the components of the fifth row of Γ.

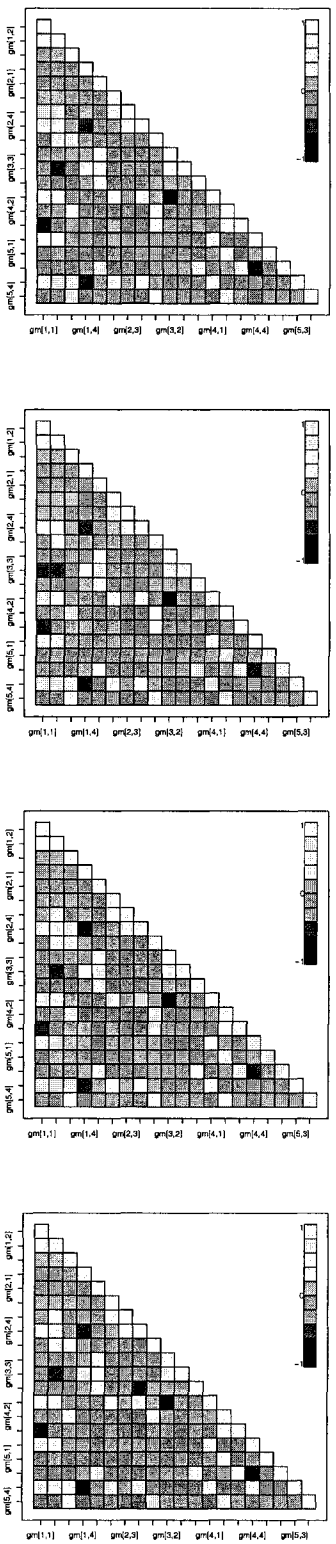

Figure C.24 Crosscorrelations among parameters by chain (chains are ordered from top to bottom).

Table C.7 Γ: Raftery and Lewis diagnostic (chain 1).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
[1, 1]	2	3802	3746	1.010
[1, 2]	2	3834	3746	1.020
[1, 3]	2	3804	3746	1.020
[1, 4]	2	3832	3746	1.020
[2, 1]	2	3834	3746	1.020
[2, 2]	2	3680	3746	0.982
[2, 3]	2	3710	3746	0.990
[2, 4]	2	3771	3746	1.010
[3, 1]	2	3815	3746	1.020
[3, 2]	2	3741	3746	0.999
[3, 3]	2	3710	3746	0.990
[3, 4]	2	3834	3746	1.020
[4, 1]	2	3680	3746	0.982
[4, 2]	2	3771	3746	1.010
[4, 3]	2	3771	3746	1.010
[4, 4]	1	3740	3746	0.998
[5, 1]	2	3741	3746	0.999
[5, 2]	2	3710	3746	0.990
[5, 3]	2	3650	3746	0.974
[5, 4]	2	3771	3746	1.010

$[i, j]$ represents the element of the ith row and jth column

Table C.8 Γ: Raftery and Lewis diagnostic (chain 2).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3741	3746	0.999
$[1,2]$	2	3771	3746	1.010
$[1,3]$	2	3741	3746	0.999
$[1,4]$	2	3771	3746	1.010
$[2,1]$	2	3710	3746	0.990
$[2,2]$	2	3741	3746	0.999
$[2,3]$	2	3802	3746	1.010
$[2,4]$	2	3650	3746	0.974
$[3,1]$	2	3834	3746	1.020
$[3,2]$	2	3802	3746	1.010
$[3,3]$	2	3802	3746	1.010
$[3,4]$	2	3620	3746	0.966
$[4,1]$	2	3650	3746	0.974
$[4,2]$	2	3680	3746	0.982
$[4,3]$	2	3771	3746	1.010
$[4,4]$	2	3680	3746	0.982
$[5,1]$	2	3865	3746	1.030
$[5,2]$	2	3680	3746	0.982
$[5,3]$	2	3741	3746	0.999
$[5,4]$	2	3834	3746	1.020

$[i,j]$ represents the element of the ith row and jth column

Table C.9 Γ:Raftery and Lewis diagnostic (chain 3).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3897	3746	1.040
$[1,2]$	2	3865	3746	1.030
$[1,3]$	2	3741	3746	0.999
$[1,4]$	2	3680	3746	0.982
$[2,1]$	2	3835	3746	1.020
$[2,2]$	2	3802	3746	1.010
$[2,3]$	2	3771	3746	1.010
$[2,4]$	2	3771	3746	1.010
$[3,1]$	2	3802	3746	1.010
$[3,2]$	2	3771	3746	1.010
$[3,3]$	2	3771	3746	1.010
$[3,4]$	2	3711	3746	0.991
$[4,1]$	2	3741	3746	0.999
$[4,2]$	2	3771	3746	1.010
$[4,3]$	2	3771	3746	1.010
$[4,4]$	2	3650	3746	0.974
$[5,1]$	2	3897	3746	1.040
$[5,2]$	2	3834	3746	1.020
$[5,3]$	2	3680	3746	0.982
$[5,4]$	2	3650	3746	0.974

$[i,j]$ represents the element of the ith row and jth column

Table C.10 Γ: Raftery and Lewis diagnostic (chain 4).

Estimand	Burn-in (M)	Total (N)	Lower bound (Nmin)	Dependence factor (I)
$[1,1]$	2	3771	3746	1.010
$[1,2]$	2	3710	3746	0.990
$[1,3]$	2	3847	3746	1.030
$[1,4]$	2	3771	3746	1.010
$[2,1]$	2	3772	3746	1.010
$[2,2]$	2	3680	3746	0.982
$[2,3]$	2	3620	3746	0.966
$[2,4]$	2	3741	3746	0.999
$[3,1]$	2	3834	3746	1.020
$[3,2]$	2	3741	3746	0.999
$[3,3]$	2	3710	3746	0.990
$[3,4]$	2	3834	3746	1.020
$[4,1]$	2	3680	3746	0.982
$[4,2]$	2	3741	3746	0.999
$[4,3]$	2	3865	3746	1.030
$[4,4]$	2	3741	3746	0.999
$[5,1]$	2	3834	3746	1.020
$[5,2]$	2	3802	3746	1.010
$[5,3]$	2	3680	3746	0.982
$[5,4]$	2	3590	3746	0.958

$[i,j]$ represents the element of the ith row and jth column

BIBLIOGRAPHY

[1] Amit, Y., and Piccioni, M. (1991). A non-homogeneous Markov process for the estimation of Gaussian random fields with nonlinear observations. *Ann. Probab.* **19**:1664-1678.

[2] Arfken, G. (1985). *Mathematical Methods for Physicists.* Academic Press, Orlando.

[3] Bagchi, P. and Guttman, I. (1988). Theoretical considerations of the multivariate von Mises-Fisher distribution. *J. of Appl. Stat.*, **15**:149-169.

[4] Banfield, J.D. and Raftery, A.E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, **4**:803-821.

[5] Berger, J.O. (1985). *Statistical Decision Theory and Bayesian Analysis*, second edition. Springer, New York.

[6] Boothby, W.M. (1975). *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, New York.

[7] Carriquiry, A.L., Stern. H.S., and Daniels, M. (2000). Statistical Treatment of Class Evidence: Trace Element Concentration in Bullet Lead. Research report submit-

ted to the Federal Bureau of Investigations. Department of Statistics, Iowa State University, Ames, IA.

[8] Casella, G. and Berger, R.L. (2002). *Statistical Inference*, second edition. Duxbury, California.

[9] Chikuse, Y. (2003). *Statistics on Special Manifolds.* Springer, New York.

[10] Cowles, M.K. and Carlin, B.P. (1995). Markov Chain Monte Carlo diagnostics: A comparative review. *J. Amer. Stat. Soc.*, **91**:883-904.

[11] Downs, Thomas D. (1972) Orientation statistics. *Biometrika*, **59**:665-676.

[12] Fisher, N.I., Lewis, T. and Embleton B.J.J. (1987). *Statistical Analysis of Spherical Data*, Cambridge University Press, Cambridge.

[13] Fisher, N.I. (1996). *Statistical Analysis of Circular Data*, 2nd. ed. Cambridge University Press, Cambridge.

[14] Fokoue, E., and Titterington, D.M. (2003). Mixtures of factor analyzers - Bayesian estimation and inference by stochastic simulation. *Machine Learning* **50**:73-94.

[15] Fraley, C. and Raftery, A.E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. Technical report No. 329, Department of Statistics, University of Washington , Seattle, WA.

[16] Fraley, C. and Raftery, A.E. (2002). Model-based clustering, discriminant analysis, and density estimation. *J. Am. Statist. Assoc.*, **97**:611-631.

[17] Gelfand, A.E. and Smith, A.F.M. (1990). Sampling based approaches to calculating marginal densities. *J. Am. Statist. Assoc.*, **85**:398-409.

[18] Gelman, A. and Rubin., D.B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statist. Science*, **7**:457-511.

[19] Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (2003). *Bayesian Data Analysis*, 2nd. ed. Chapman&Hall-CRC Press, New York.

[20] Halmos, P.R. (1974). *Measure Theory.* Springer, New York.

[21] Harville, D.A. (1997) *Matrix Algebra from a Statistician's Perspective.* Springer, New York.

[22] Helgason, S. (1978). *Differential Geometry, Lie Groups, and Symmetric Spaces.* Academic Press, Orlando.

[23] Hoff, P.D. (2006). Model averaging and dimension selection for the singular value decomposition. Working paper No. 54. Center for Statistics and the Social Sciences, University of Washington, Seattle, WA.

[24] Hoffman, D.K., Raffenetti, R.C., and Ruedenberg, K. (1972). Generalization of Euler angles to $N-$dimensional orthogonal matrices. *J. Math. Phys.*, **13**: 528-533.

[25] James, A.T. (1954). Normal multivariate analysis and the orthogonal group. *Ann. Math. Statist.*, **25**:40-75.

[26] James, A.T. (1964). Distributions of matrix variates and latent roots derived from normal samples. *Ann. Math. Statist.*, **27**: 475-501.

[27] Johnson, R.A. and Wichern, D.W. (2002) *Applied Multivariate Statistical Analysis.* 5th. ed. Prentice Hall, New Jersey.

[28] Jupp, Peter E. and Mardia, Kanti V. (2000). Maximum likelihood estimators for the matrix von Mises-Fisher and Bingham distributions. *Ann. Statist.* **7**:599-606.

[29] Lancaster, P. and Tismenetsky, M. (1985). *The Theory of Matrices.* Academic Press, New York.

[30] Jupp, Peter E. and Mardia, Kanti V. (2000). *Directional Statistics.* Wiley, New York.

[31] Khatri, C.G. and Mardia, K.V. (1977). The von Mises-Fisher distribution in orientation statistics. *J. Roy. Statist. Soc, Ser. B*, **39**: 95-106.

[32] Kim, P.T. (1991). Decision theoretic analysis of spherical regression. *J. Multivariate Anal.*, **38**:233-240.

[33] Mardia, K.V. and El-Atoum, S.A.M. (1976). Bayesian inference for the von Mises-Fisher distribution. *Biometrika*, **63**:203-206.

[34] Mathai, A.M., Provost, S.B, and Hayakawa, T. (1995). *Bilinear Forms and Zonal Polynomials.* Springer, New York.

[35] Peele, E.R., Havekost, D.G., Peters, C.A., Riley, J.P., Halberstam, R.C., and Koons, R.D. (1991). Comparison of bullets using the elemental composition of the lead component. *Proceedings of the International Symposium on the Forensic Aspects of Trace Evidence*, pp. 57-68.

[36] Raftery, A.E. and Lewis, S. (1992). How many iterations in the Gibbs sampler?. In *Bayesian Statistics 4* pp 763-773. Oxford University Press, Oxford.

[37] Robert C.P. and Casella, G. (2004). *Monte Carlo Statistical Methods.* Springer, New York.

[38] Slater, L.J. (1966). *Generalized Hypergeometric Functions.* Cambridge University Press, Cambridge.

[39] Swayne, D.F., Cook, D., Bujas, A. (1998). XGobi: Interactive dynamic data visualization in the X Window system. *J. Comp. Graph. Statist.* **7**:113-130.

[40] Ward, J.H. (1963). Hierarchical grouping to optimize an objective function. *J. Am. Statist. Assoc.*, **58**:236-244.

[41] Warner, F.W. (1983). *Foundations of Differentiable Manifolds and Lie Groups.* Springer-Verlag, New York.

[42] Watson, G.S. (1983). *Statistics on Spheres.* Wiley, New York.